Lossy Compression of Quality Values in Sequencing Data

Veronica Suaste Morales and Sheridan Houghten, *Member, IEEE*

Abstract—The dropping cost of sequencing human DNA has allowed for fast development of several projects around the world generating huge amounts of DNA sequencing data. This deluge of data has run up against limited storage space, a problem that researchers are trying to solve through compression techniques.
In this study we address the compression of SAM files, the standard output files for DNA alignment. We specifically study lossy compression techniques used for quality values reported in the SAM file and analyze the impact of such lossy techniques on the CRAM format. We present a series of experiments using a data set corresponding to individual NA12878 with three different fold coverages. We introduce a new lossy model, dynamic binning, and compare its performance to other lossy techniques, namely Illumina binning, LEON and QVZ. We analyze the compression ratio when using CRAM and also study the impact of the lossy techniques on SNP calling. Our results show that lossy techniques allow a better CRAM compression ratio. Furthermore, we show that SNP calling performance is not negatively affected and may even be boosted.

Index Terms—Compression, next-generation sequencing, quality values, SNP calling performance.

1 Introduction

It is expected that by the year 2025 two billion human genomes will be sequenced [33]. This deluge of data represents a huge challenge in terms of storage space. To help address this issue, researchers have studied several compression techniques for such data, varying from general text compression techniques to specialized models that exploit particular properties of DNA strands.

Next-Generation Sequencing (NGS) technologies include *base-calling* algorithms that infer the actual nucleotide information and then assign a measure of uncertainty (*quality score*) to each base call. These vary depending on the sequencing system. Following base-calling and alignment, the next step is *SNP calling*, or *variant calling*, which determines where polymorphisms exist or where there is a difference from a reference sequence. This is then followed by *genotype calling*, which determines the genotype for each individual and is closely related to the position of a SNP or variant that has already been called.

Our main target for this study is the quality values reported in FASTQ files, the standard format for storing the output of high-throughput sequencing instruments, as well as SAM files, the standard format for storing read alignments against reference sequences. These quality values report a score per base associated with the nucleotide sequence and represent the probability of an error in base calling. The alphabet used for quality scores consists of some 40 characters, presenting another barrier for achieving good compression.

A compression algorithm takes input X and generates a representation X_c that requires fewer bits. The inverse process operates on the compressed representation X_c to generate the reconstruction, Y [32]. Data compression schemes are *lossless* if $Y = X$ or *lossy* if Y is only an approximation for X; in the case of lossy compression, it is not expected that the original data can be recovered exactly.

Recently, new lossy models of compression for quality values have been studied, e.g. LEON [3], QVZ [24] and Illumina binning [19]. In this study, we analyze these three, and also introduce new ideas for adjusting quality scores using *dynamic binning*.

Our analysis is related to the CRAM format, which compresses SAM/BAM files to achieve 40-50% space saving over BAM, which in turn achieves 50-80% space saving over SAM. An objective of this format is to replace BAM and become the standard compression model for sequencing data.

We will study the impact of the lossy models in CRAM. When using lossy compression one must study the effect of the loss of information on subsequent tasks performed with compressed data. Hence, with respect to lossy compression of quality values, we will study the effect of the compression on SNP calling. Some recent results [39] suggest SNP calling performance is not negatively affected and can even be boosted.

The remainder of this paper is structured as follows. Section 2 provides an overview of data compression in genomics, as well as information on the formats considered (SAM, BAM and CRAM). Section 3 presents the methodology, and includes information on the datasets, the performance metrics, the new dynamic binning technique, and the process followed to apply each of the lossy compression techniques to the CRAM format. Sections 4, 5 and 6 present the results and analysis for each of the three datasets considered. Section 7 presents conclusions and discusses possible next steps.

- *V. Suaste Morales was with the Department of Computer Science, Brock University, St. Catharines, ON, Canada. She is now with CONABIO, Mexico. E-mail: vsuaste@conabio.gob.mx*
- *Sheridan Houghten is with the Department of Computer Science, Brock University, St. Catharines, ON, Canada. E-mail: shoughten@brocku.ca*

2 Data Compression in Genomics

The first approaches for compression of sequencing data were based on text compression techniques adapted to exploit obvious properties of DNA sequences such as the 4-letter alphabet, regularities and presence of palindromes [15]. All these techniques essentially used a combination of two different methods: *dictionary based*, in which most repetitive subsequences are identified and encoded with a representation of smaller size, and *statistical based*, in which a prediction model is established which assigns probabilities to each base based on the data and then uses an encoding scheme that will perform more efficiently based on the probability distribution.

With advances from NGS technologies, new challenges also emerged for compression of its output data because these technologies, along with the sequence or read itself, also report additional metadata needed for downstream DNA analysis. This metadata uses a larger alphabet than the 4-letter alphabet that had been considered for sequencing data compression until that time. We first introduce the data formats which are the target for compression, and then give an overview of the research that has been developed.

2.1 Data Formats

2.1.1 FASTQ Format [8]

This has become the standard format for storing output of high-throughput sequencing instruments. It stores the nucleotide sequence and its corresponding quality scores, each encoded with a single ASCII character.

2.1.2 Sequence Alignment/Map format (SAM) [31]

This is a generic format for storing read alignments against reference sequences, developed with the main purpose of allowing DNA analysis and the exchange of information from various sequencing platforms. BAM format, the binary version of SAM, is designed to compress reasonably well. These two formats are industry standards for reporting alignment/mapping information. Also all the important tools for analysis of high-throughput sequencing data require these formats as the input, including for example GATK [25] and Samtools [22].

2.2 DNA Sequence Compression

Although compression of the DNA sequences themselves is not the target of the current study, for completeness we provide a brief overview in this section. Since DNA strings contain only four possible symbols (A, C, G, T) many repetitions are expected, and this property has been broadly exploited for DNA sequencing compression techniques.

Reference-based methods first choose a reference sequence and then only encode the differences between it and the sequence to be compressed. For good performance the choice of the reference is important, and in some cases a set of possible references is allowed [2], [4], [11]. Other approaches focus only on output of NGS technologies, e.g. Fastqz [6] for FASTQ format and mzip [17] for SAM format.

De novo or *reference-free* compression is performed without an external reference genome, instead exploiting similarities between the reads themselves. Most commonly these techniques use a context-model to predict the bases and then an arithmetic encoder, or they will re-order reads to maximize similarities for consecutive reads allowing a better compression with standard methods. According to [3], read re-ordering methods are the ones that achieve a better compression ratio. Compressors of this type include BEETL [20], ORCOM [13], MINCE [28], and LEON [3].

2.3 Random Access and CRAM Format

Although compression mainly focuses on the issue of storage and distribution of big data, it is also possible to make more practical the step of analyzing the data, by applying particular compression techniques for specific files. This allows access to part of the file, from its compressed version, without going through the entire decompression process. For achieving random access compression, generally the input is split into blocks and different compression algorithms may even be applied to different blocks within the same file.

Today, BAM [31] is the standard compression model with random access property, achieving compressions of 50-80% of the original SAM file and allowing accessible and practical analysis of sequencing data using the compressed file. However, this compression ratio is not sustainable in the long run due to the huge growth in sequencing data. Because of this, researchers have explored new options.

One of the best is CRAM [9], which is based on [17] and compresses SAM/BAM files achieving 40-50% space saving over BAM. It is now supported for the main tools for analysis of sequencing data and also big initiatives such as the 1000 genomes project have their data available for public use stored in this format. After aligning the sequence data to a reference, CRAM stores only the data that is different rather than the entire genome. Also, different compression algorithms are used for each data type such as sequence names, genetic sequences and quality values [10].

2.4 Quality Values

The base calling process performed for NGS technologies to determine the bases of a DNA string is prone to errors. To report the probability of base calling mistakes, sequencers generate a quality score for each nucleotide in the read, indicating the level of confidence of a particular read base. The higher the score, the lower the probability that the base at that position has been incorrectly called. Usually the quality values are represented in a file with a printable ASCII alphabet [33:73] or [64:104]. The importance of maintaining quality scores as part of the data relies on the fact that they are directly used in next-generation sequencing analysis, such as Single Nucleotide Polymorphism(SNP) detection [12]. Quality scores comprise a significant percentage of sequencing data and are a bottle neck for compression because the alphabet required to represent them is larger – about 40 characters – than that required for the read sequence. Compression algorithms designed for the sequence itself will thus not perform as well when applied to quality values, and so specific algorithms must be provided that take into account the properties and information from the quality scores.

TABLE 1
Quality Score Bins for Illumina Binning

Quality Score Bins	Mapped Quality Score
N(no call)	N(no call)
2-9	6
10-19	15
20-24	22
25-29	27
30-34	33
35-39	37
≥ 40	40

2.5 Lossy Compression of Quality Values

Lossy compression techniques are not suitable for the DNA sequences themselves, as losing or changing a single base has a big impact on the possible encoded protein. However researchers have applied lossy compression techniques to the quality values, which represent almost half of the total data in the output file. There are several examples of quality values compression using lossy techniques, e.g. [7], [3], [38], [24], and [26].

When lossy compression is applied to any type of data, it is important to see the effect of the lost data on any other output derived from the compressed data. Any lossy technique in this field must consider the effects on downstream analysis, SNP calling. In [26] it is suggested that quality values data is noisy, and their analysis showed that losing precision in quality scores was actually beneficial for SNP calling.

A number of lossy techniques have been proposed for compression of quality values. In [7] a lossy technique is proposed in which quality values are separated into blocks of variable size, with each block having one representative value. Two main options are used, *P-block* which is controlled by parameter p and which uses Mean Manhattan Distance, and *R-block* which has a maximum ratio r and makes use of varying levels of precision depending on whether the quality score is higher or lower.

Concurrently, Illumina [19] suggested what is now known as *Illumina binning* [19], which reduces the resolution of quality scores by using at most 8 levels of quality, as shown in Table 1. QualComp [26] allows users to specify the number of bits per quality score prior to compression. *Quality Values Zip* (QVZ) [24] models the quality score sequence as a Markov chain of order one, while LEON [3] builds a de Bruijn Graph of the most recurrent *k-mers* in the sequences, with each read encoded as a path in the graph; for quality score compression all quality values above a threshold are truncated and also all positions covered by at least a certain number of recurrent *k-mers* are replaced by one representative based on quality value.

A recently-proposed technique is CALQ [37], which specifically considers downstream analysis in its decisions, making use of alignment information to determine an acceptable level of distortion of the quality values to ensure downstream analysis is not negatively affected. Another recent technique is Crumble [5], which uses various heuristics to identify which quality values are necessary based on whether the lack of those values will have a negative effect on variant calling.

2.6 Comparison and Discussion

Lossy compression of quality values is a recently explored area, with the first ideas presented in 2011 [17]. When comparing lossy compression techniques, many factors are involved and there is a trade off between compression ratio and other measures, e.g. distortion rate, impact on downstream analysis, speed or memory requirements. Many different metrics are used to measure distortion rate, see e.g. [7], [24] and [26]. Impact on downstream analysis (SNP calling) is normally presented using F-score which considers both precision and recall measures; see Section 3.4.

Qualcomp [26] focusses its comparisons and performance measurement on rate distortion metric, in particular mean square error. They study SNP calling but we note that they consider as ground truth the data resulting from variant calling performed with the original quality values. The running time is longer than their comparators [16], [6] but they achieve better compression ratio minimizing mean square error with little compromise in SNP calling. In comparison, [7] used fidelity measures and outperformed Qualcomp considering *Max : Min Distance* as the measure; they also based their SNP analysis considering variant calling with original quality values and did not report data for running time and memory storage.

In [24] the performance of QVZ is compared to [26] and [7], outperforming both for all three choices of distortion metric. A later paper [27] presents an exhaustive analysis on the effect of lossy compression of quality values using QVZ on variant calling. This analysis used two different benchmarks, one by Genome in a Bottle and adapted by the National Institute of Standardizations and Technology, and the other by Illumina as part of the Platinum Genomes project. They also included Illumina binning in their comparison. Not only is storage space reduced, but they also show that SNP calling is not negatively affected. Moreover, they confirm with their experiments that smoothing quality values can improve SNP calling performance.

LEON [3] is based on a probabilistic de Bruijn graph designed originally for compression of the sequence; the graph is then also used to perform a lossy transformation of quality values. We must keep this in mind because the graph has high memory requirements. Their analysis showed better performance by LEON for compression ratio, compression time and decompression time compared to [6], [16], [28]. They also presented SNP calling analysis considering the benchmarks provided by 1000 genomes project and also showed that using lossy techniques on quality values can improve SNP calling.

2.7 Objectives of this Study

In this work we address the compression of SAM files, the standard output for DNA alignment. Specifically, we consider lossy compression techniques for quality values reported in SAM.

There are multiple objectives for this work. We introduce a new lossy model, dynamic binning. Our first objective is to analyze its performance in comparison to three of the most promising lossy techniques: QVZ [24], LEON [3], and Illumina binning [19].

A second objective is to analyze the performance of each of these different approaches when using CRAM, which is becoming a standard in place of BAM. Because we are analysing lossy techniques, we study not only the compression ratio, but also the impact of using such methods on the SNP calling performance.

To address these objectives we use 3 datasets of varying coverage as well as 2 different callers, thereby assisting us in observing trends. Following previous studies such as [27] we concentrate our analysis on chromosomes 11 and 20, however results for the complete genome are also presented for one of the datasets.

3 METHODOLOGY

In this section we present the toolkits, software and data sets used for our research. We also present the metrics for evaluating SNP calling performance, and explain the experimental process followed in our work.

3.1 Other File Types and Toolkits

3.1.1 Variant Call Format (VCF) [36]

This is a text file format for storing gene sequence variations, and is the output of variant calling performed by GATK tools or Samtools. It includes a quality score for each alternative allele it lists.

3.1.2 Genome Analysis Toolkit (GATK) [25]

This is a collection of command-line tools for analyzing high-throughput sequencing data in formats such as SAM/BAM/CRAM and VCF with a primary focus on variant discovery and genotyping. We follow GATK Best Practices [12], [35], which aims to maximize the technical correctness of the data. The first steps start from the raw reads indicating how to do the mapping to a reference genome, marking duplicates with Picard tools (see subsection 3.1.5) and performing a base quality score recalibration. Once the data has been pre-processed it is ready to continue with the variant discovery process. In this second part the process considers the fact that some of the variation might be caused by mapping and sequencing artifacts. Finding a good trade-off between sensitivity (minimizing false negatives) and specificity (minimizing false positives) can be very difficult, and can also be dependent on the project. Instead, the process maximizes sensitivity but also reports a variant quality score recalibration (VQSR) which further allows the user to customize specificity for each project.

3.1.3 Samtools [22]

This is a suite of programs for interacting with sequencing data in SAM/BAM/CRAM formats, allowing one to work directly with a compressed BAM/CRAM file. We also use this toolkit for SNP calling via the *mpileup* command, which calculates genotype likelihoods supported by aligned reads and does SNP calling based on those likelihoods. We use Bcftools, another module, to handle VCF files.

3.1.4 HTSlib

This is a C library for manipulating file formats (e.g. SAM/CRAM/VCF). It is used to study high-throughput sequencing data and is the core library used by Samtools.

3.1.5 Picard [29]

This is a set of command line tools for manipulating high-throughput sequencing data. We use it for marking duplicate reads as part of the GATK Best Practices.

3.1.6 Burrows-Wheeler Aligner (BWA) [21]

This is software for mapping low-divergent sequences against a large reference genome. When using this to perform alignments we used reference genome GRCh37 (available at [14]), the reference genome used in phase 1 and phase 3 of 1000 Genomes Project.

3.2 Datasets For SNP Calling

To analyze the impact of lossy compression models for quality values we use datasets from 1000 Genomes project public repository [1], all corresponding to *Homo Sapiens* individual NA12878, the daughter in one of the trios sequenced from Utah residents of northern and western European ancestry (CEU). We used NA12878 because it is the only one for which a well analyzed ground truth set of variants has been developed and publicly released.

We first consider a complete process experiment, using the read dataset SRR622461 for individual NA12878 with 5x coverage, performing alignment on this raw dataset using BWA [21] and then extracting chromosomes 11 and 20. We then perform experiments with the whole genome, using low coverage alignment (6x). Finally, we consider high coverage alignment (50x) but for chromosomes 11 and 20 only.

3.3 Quality Benchmarks for SNP Calling

To measure how lossy models affect SNP calling, we must set the baseline that will serve as a reference when comparing the performance of lossless compression against the different lossy models. For this purpose we use two *ground truth* sets of variants developed and refined specifically for individual NA12878.

The first of these was released by the Genome in a Bottle consortium (GIAB) [40], and has been adapted by the National Institute of Standardizations and Technology (NIST). They integrated and arbitrated between 14 data sets from five sequencing technologies, seven read mappers and three different variant callers, resulting in a set of variants allowing for high-confidence SNP calling without depending on specific caller or sequencing technologies.

The second gold standard we used is the one released by Illumina as part of the Platinum Genomes project [30]. It includes a set of high-confidence variant calls for individual NA12878.

3.4 SNP Calling Performance Metrics

We will evaluate how different lossy models for quality values affect SNP calling. We consider this problem as a binary classifier for the variants, identifying a variant in the resulting VCF file as True Positive (TP) when it is also part of the ground truth and False Positive (FP) when it can not be found in the ground truth. False Negatives (FN) are the variants in the ground truth that cannot be found in the

TABLE 2
Quality Score Bins for Dynamic Binning

Quality Score Bins	Mapped Quality Score
1-10	$c_1, H(c_1) \geq H(c)\ \forall c \in [1, 10]$
11-20	$c_2, H(c_2) \geq H(c)\ \forall c \in [11, 20]$
21-30	$c_3, H(c_3) \geq H(c)\ \forall c \in [21, 30]$
31-40	$c_4, H(c_4) \geq H(c)\ \forall c \in [31, 40]$
41-256	$c_5, H(c_5) \geq H(c)\ \forall c \in [41, 256]$

resulting VCF file. The performance will be evaluated by the following:

1) **Sensitivity** measures the proportion of correctly identified positives: *Sensitivity* $= \frac{TP}{TP+FN}$
2) **Precision** measures the proportion of identified positives that are true: *Precision* $= \frac{TP}{TP+FP}$
3) **F-score** considers both precision and sensitivity. It can be interpreted as a weighted average and is computed as: *F-score* $= 2 \times \frac{Sensitivity \times Precision}{Sensitivity + Precision}$

Sensitivity is also known as the *true positive rate*, while the *false positive rate* is calculated as $\frac{FP}{FP+TN}$.

When analyzing results given by these metrics note that a perfect sensitivity score of 1.0 indicates that all variants from the ground truth were correctly identified but does not indicate how many irrelevant variants were also called as positive. Meanwhile, a precision score of 1.0 means all obtained variants are relevant but indicates nothing about the set of possible positive variants. Generally there is a trade-off between these two metrics and depending on the case one may prefer to increase one of them by decreasing the other. Because of this, F-score, which combines both metrics, gives us a better overall evaluation.

The *Receiver Operating Characteristic (ROC) Curve* is used to visualize the performance of a binary classifier while varying a threshold. It is the result of plotting the true positive rate against the false positive rate with different thresholds.

When evaluating variant calling performance with sensitivity, precision and F-score, we considered all variants in an output VCF file to be correct. Using ROC as metric, we vary the quality threshold and consider variants in an output VCF file to be correct only if they are above the threshold.

This metric, specific to variant calling performance, was introduced in [39]. Following their design, when comparing different sets of variants we take their union as the domain. This rescaling is done to address the fact that the true negative rate of correctly called variants is so much larger, as most of the genome is variant, and so it could cause misleading results. A result of this rescaling is that ROC curves between different plots are not comparable as they have different domains. For our analysis we also look at the area under the curve (AUC), which gives the probability that the binary classifier will rank a random positive case higher than a random negative case. The closer AUC is to 1, the better.

3.5 Dynamic Binning

To explore new ideas for reducing the alphabet used for quality values, we developed *dynamic binning*. As in Illumina binning, this method splits the alphabet into bins. However, while Illumina binning uses 8 bins, dynamic binning uses 5 bins and each bin is represented by the value which has the most occurrences in that bin. We apply this method block-wise: the file is split into blocks and for each block the 5-binning depends on its histogram. In our experiment we used blocks of 1000 reads, an empirically selected parameter.

For a given block, let H denote the histogram of the quality values: i.e. for character c, $H(c)$ is the number of occurrences of c in the block. The 5-binning for each block is performed according to Table 2 where for each bin its representative value c_i satisfies $H(c_i) \geq H(c)$ for all values c in the range covered by the bin. In such a way each entry within a given bin is represented by the most common entry within that bin. Figure 1 shows an example of a histogram for chromosome 20, 5x coverage, with representative values of each bin shown in red.

Note that different blocks within the file will likely have different histograms, as frequencies of quality values are likely to change from one block to another. As a result, the representative values for the bins are expected to vary from one part of the file to another. Although dynamic binning was inspired by Illumina binning, it therefore allows for a more accurate representation of the quality values within each block as quality values vary across the file. Note that the ranges covered by the bins do not vary across the file, although this is a possible avenue for future research (see Section 7).

Although the number of bins used by dynamic binning is smaller than Illumina's traditional set of 8 bins, it will be shown in Sections 4–6 that the use of dynamic rather than fixed values means that dynamic binning is very competitive in terms of downstream analysis while also improving the compression ratio.

3.6 Process

We study how CRAM compression can be improved by modifying quality values with four different techniques: our new technique dynamic binning, as well as QVZ [24], LEON [3], and Illumina binning [19]. We also analyze how the use of these lossy techniques impacts SNP calling.

To perform CRAM compression we need as input a SAM file with the quality values modified by each of the techniques to analyze. The general work flow has the following steps:

1) We first align the reads to reference genome GRCh37. The output is a SAM file that we call the *original* or *raw* SAM, with quality values called the *original* or *raw* quality scores.
2) We next create a SAM file for each of the lossy models, with the quality scores updated according to each model, and then convert each SAM file to CRAM format (v3.0) using Samtools.
    - For dynamic binning: compute the histogram for each block and create a dictionary of representative values for each bin per block, then apply the transformation to quality values according to the dictionary.
    - For Illumina binning: modify the code in *htslib* so when converting the BAM file to CRAM

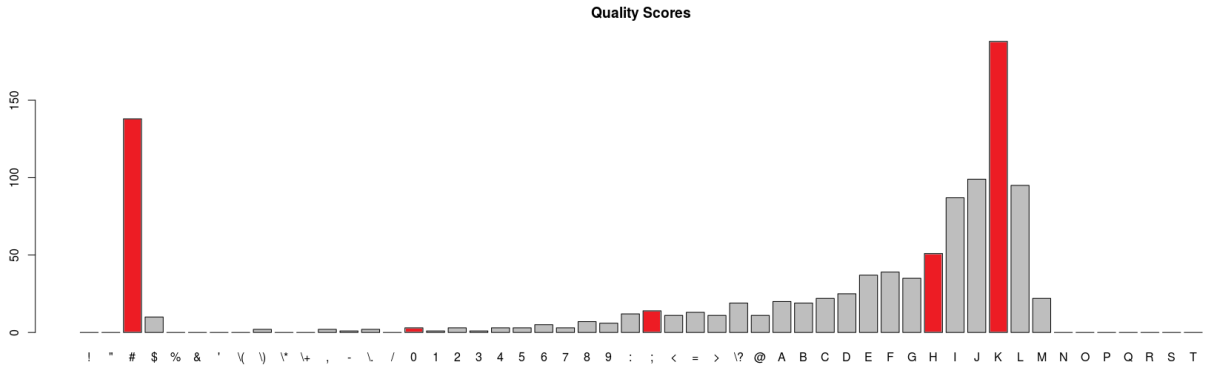

Fig. 1. Quality values histogram, chromosome 20. Red values are the representatives of each bin.

it transforms each quality value according to Table 1.

- For LEON: create a temporal FASTQ file with the reads and quality values extracted from the *original* SAM. Run LEON algorithm with this created file, then decompress the output and create the new SAM file with the modified quality values.
- QVZ: Extract the quality values from the *original* SAM file. Once the quality values are processed decompress the output and create a SAM file with the new scores. In our experiments, for QVZ the ratio was set to 0.8 and we used default values for all other parameters.

3) We compare the compression ratio, as well as the impact of each of the techniques on SNP calling performance, following *Best Practices* [12], [35] to improve the data. We perform the variant calling with two different tools (Samtools and GATK) and compare sensitivity, precision and F-score for each of the lossy models. In this step, from Samtools we use *samtools mpileup* pipeline and from GATK toolkit we perform variant calling with *Haplotype Caller* command.

4 Results and Analysis – Complete Process Experiment (5x Coverage)

This section examines the dataset of reads SRR622461, corresponding to individual NA12878 with 5x fold coverage. From performing the alignment with BWA we obtained a SAM file of size 49GB, from which we extracted chromosomes 11 (2.3GB) and 20 (1.0GB). For each of these we applied the four lossy techniques to the quality values, converted each output to SAM and then converted the resulting file to CRAM.

4.1 Compression Ratio

Table 3 shows a summary of file sizes for each of the lossy models as well as the raw SAM file. This table contains SAM and CRAM file size to analyse the compression ratio, which is included in the last row of each section.

TABLE 3
5x Fold Coverage: File Size (rounded to closest MB) and Compression Ratio

	Raw	Illumina	QVZ	LEON	Dynamic bin
Chromosome 11					
SAM	2357	2357	2357	2357	2357
BAM	452	363	445	310	341
CRAM	168	102	159	73	87
Compression	13.99	23.17	14.80	32.31	27.00
Chromosome 20					
SAM	1037	1037	1037	1037	1037
BAM	206	165	196	137	155
CRAM	77	47	70	32	40
Compression	13.40	22.24	14.75	32.04	26.09

Notice that the CRAM format by itself already performs a very good compression, as displayed in the Raw column. Without adjusting quality values at all, the SAM file is over 13 times larger than the CRAM file for both chromosomes.

All of the lossy techniques had a favorable impact on compression ratio. For both chromosomes LEON had the best impact on compression ratio, followed by dynamic binning. Both of these lossy techniques roughly doubled the compression in comparison to the raw data; for chromosome 11 the compression ratio is 13.99 for the raw data, 32.31 for LEON and 27.00 for dynamic binning, while for chromosome 20 the compression ratio is 13.40 for the raw data, 32.04 for LEON and 26.09 for dynamic binning.

Table 3 also includes BAM file size to appreciate how the lossy models for quality scores also have an impact on the BAM format. As can be seen in this table, all lossy techniques improved BAM file size, although the improvement is much more significant for CRAM. For example, in chromosome 11 the raw BAM file is 452MB in size and applying LEON reduces the BAM file size to 310MB, or roughly 70% of the raw BAM size; in comparison, applying LEON to the corresponding raw CRAM file (168MB) produces a file that is less than half its size (73MB).

4.2 Variant Calling Performance

We now consider the impact of the loss of information on variant calling performance.

When we transform the quality scores, variant calling results may be influenced by these transformations. To

TABLE 4
5x Fold Coverage: Variant (SNP) Calling Performance with Illumina Ground Truth

	GATK			Samtools		
	Sens.	Prec.	F-Score	Sens.	Prec.	F-Score
Chromosome 11						
Raw	0.7201	0.9600	0.8229	0.7476	0.9488	0.8363
Illumina	0.7228	0.9605	0.8249	0.7550	0.9472	0.8402
QVZ	0.7200	0.9601	0.8229	0.7476	0.9487	0.8362
LEON	0.7165	0.9598	0.8205	0.7486	0.9429	0.8346
Dyn. bin	0.7206	0.9587	0.8228	0.7483	0.9455	0.8354
Chromosome 20						
Raw	0.6763	0.8683	0.7604	0.7063	0.8614	0.7762
Illumina	0.6787	0.8688	0.7621	0.7139	0.8606	0.7804
QVZ	0.6514	0.8644	0.7430	0.6770	0.8578	0.7567
LEON	0.6481	0.8629	0.7402	0.6780	0.8508	0.7546
Dyn. bin	0.6775	0.8670	0.7606	0.7072	0.8586	0.7756

measure and evaluate these possible changes we performed variant calling with the raw quality values and also with each of the lossy models under study. We then computed sensitivity, precision and F-score to evaluate variant calling performance against two separate ground truth sets of variants, as shown in Table 4. This table contains the results obtained for two different callers, GATK and Samtools, to demonstrate that the results do not depend on the caller. This table shows the result of the experiment using the Illumina ground truth. For further details on the same experiment with GIAB-NIST ground truth see [34]. The relative results (for different callers and different lossy techniques) are very similar for both ground truths.

Our main metric for evaluation is F-score because it combines both sensitivity and precision. Otherwise the trade-off between sensitivity and precision may not allow us to conclude anything important. For example, for chromosome 11, LEON outperforms raw quality values in sensitivity but it is the opposite for precision. From this we can not argue one is better than the other, but F-score shows that although both are very close, for raw quality values F-score is slightly higher.

This table demonstrates that even though the compression ratio is considerably better when applying any of the lossy techniques, the overall F-score does not change drastically. In fact, in some cases variant calling performance is even slightly improved when lossy techniques are used. For example, for chromosome 20 with the GATK caller, the F-scores for dynamic binning (0.7606) and Illumina binning (0.7621) are both slightly better than that of the raw quality values (0.7604).

4.3 ROC Curve Analysis

We plotted ROC curves to study the behaviour of each lossy technique when varying the threshold at which to consider a variant correctly called. Figure 2 displays the ROC curve for chromosome 11, based on Illumina ground truth with Samtools caller. At each point the raw quality values curve is always overlapped or dominated by at least one other technique. This confirms again that changing quality values with any of these techniques does not negatively affect variant calling performance. The same curve for chromosome 20 displays similar results.

TABLE 5
AUC, 5x fold coverage

Chr.	Caller	Raw	Illumina	QVZ	LEON	Dyn. bin
11	GATK	0.5149	0.5395	0.5173	0.5223	0.5105
11	Samtools	0.6766	0.6800	0.6766	0.6664	0.6724
20	GATK	0.5007	0.5094	0.5012	0.5033	0.4983
20	Samtools	0.5965	0.6020	0.5967	0.5902	0.5958

Table 5 presents, for both chromosomes, the area under the curve (AUC) as a metric for each technique and for each caller. In all cases, all lossy techniques, as well as raw quality values, have very similar AUC, with both Illumina and QVZ having higher AUC than for raw quality values. Nonetheless, it is important to mention that the AUC results are all very close to 0.5 and therefore it can be argued that the classification is not necessarily good. This is completely reasonable because from a data set with only 5x fold coverage it is not expected to have high accuracy, not even for the alignment and therefore neither for the variants. In Section 6 we show the same results for high coverage (50x) and there we can notice the differences.

5 Results and Analysis – Whole Genome Experiment (6x Coverage)

In this section of experiments we used the low coverage (6x) alignment provided by 1000 Genomes Project [1] as a 16.2GB BAM file for the whole genome, containing the alignment performed with BWA. This data set also corresponds to the individual NA12878. As in the earlier experiments, we apply the four lossy techniques to each SAM file and analyze their behaviour when converted to CRAM format, including the compression ratio and any possible impact on variant calling performance. However in this case we run the experiment with the first 22 chromosomes to get better insights with no bias to any specific chromosome. Because of storage constraints and to have greater control of the results, the experiments were performed for each chromosome separately. In this paper we present a summary of the results for all of these chromosomes, with the full results available in [34]. For comparison purposes in this paper we concentrate our analysis on results for the chromosomes used in the other experiments, namely chromosomes 11 and 20.

5.1 Compression Ratio

Table 6 summarizes the CRAM compression ratios for all 22 chromosomes. As shown in this table, similar results are seen across all chromosomes. The total size of the SAM files for all of these chromosomes is 66.11GB. After converting to CRAM, the total file size is 7.68GB for the raw quality values, giving a compression ratio of 8.61. After applying each of the lossy techniques and then converting to CRAM, the total file sizes are 4.31GB for Illumina (ratio = 15.34), 7.49GB for QVZ (ratio = 8.83), 2.58GB for LEON (ratio = 25.62), and 3.61GB for dynamic binning (ratio = 18.31).

Table 7 summarizes file sizes for chromosomes 11 and 20 in greater detail, showing SAM, BAM and CRAM sizes along with the compression ratio. The results show that every lossy technique improves compression in the BAM file for these chromosomes in comparison to the raw data.

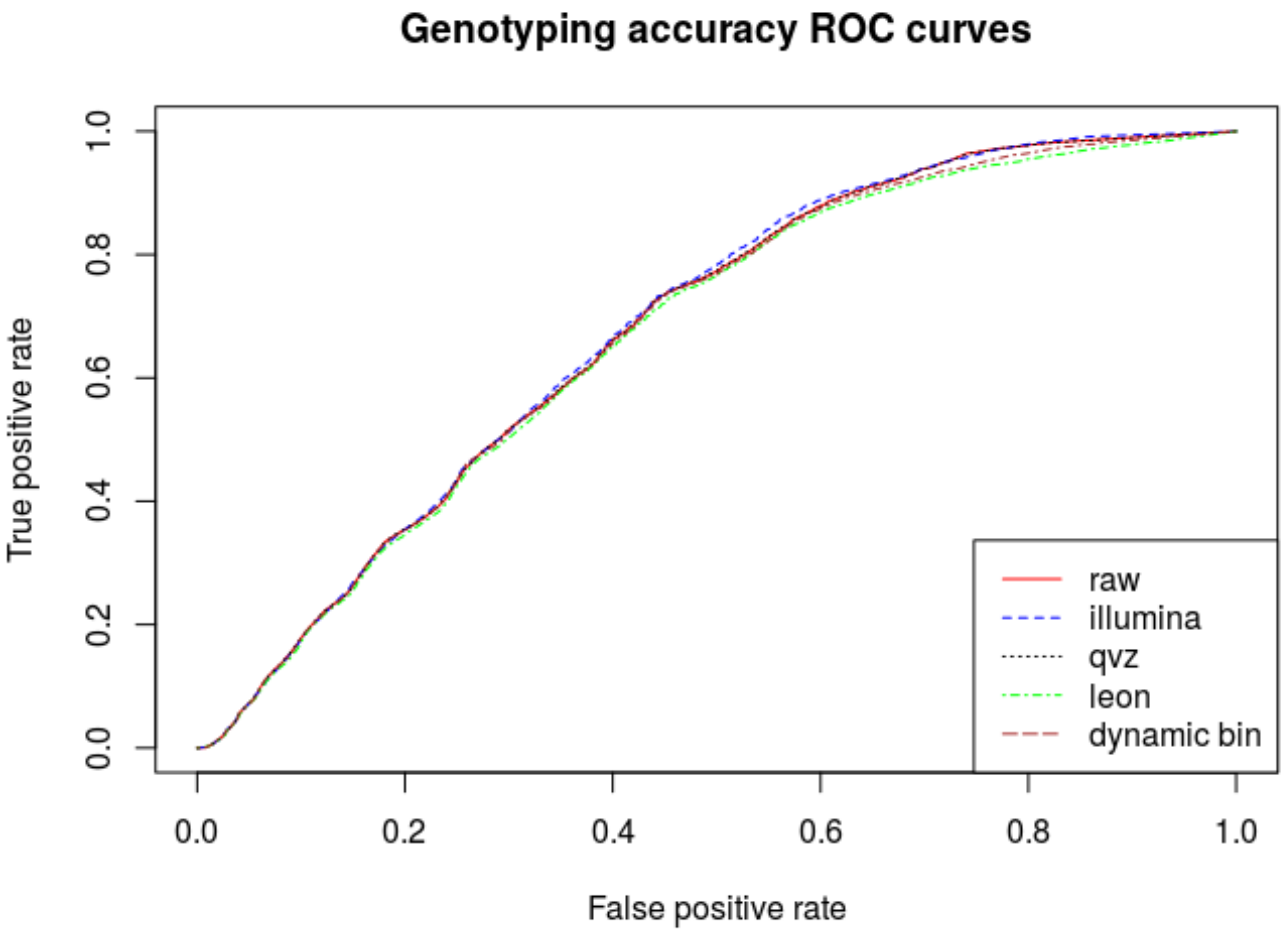

Fig. 2. ROC, chromosome 11 (5x fold coverage). True positive and false positive rates are as defined in Section 3.4 and relate to SNP calling only.

TABLE 6
6x Fold Coverage: CRAM Compression Ratio for Each Chromosome

Chromosome	Raw	Illumina	QVZ	LEON	Dynamic bin
1	8.57	15.98	8.75	25.31	18.03
2	8.64	16.33	8.83	26.13	18.44
3	8.68	16.53	8.86	26.61	18.67
4	8.70	16.69	8.88	27.00	18.84
5	8.67	16.50	8.86	26.54	18.62
6	8.66	8.85	8.85	26.49	18.60
7	8.60	16.12	8.79	25.62	18.19
8	8.65	16.38	8.84	26.27	18.50
9	8.55	15.89	8.74	25.02	17.88
10	8.61	16.08	8.79	25.42	18.16
11	8.33	15.51	9.30	24.60	19.08
12	8.64	16.25	8.82	25.90	18.36
13	8.71	16.67	8.89	26.88	18.84
14	8.63	16.23	8.82	25.79	18.33
15	8.56	15.87	8.74	25.02	17.92
16	8.44	15.29	8.65	23.80	17.22
17	8.45	15.19	8.61	23.29	17.16
18	8.66	16.45	8.84	26.33	18.59
19	8.33	14.56	8.49	21.76	16.43
20	8.55	15.76	8.72	24.51	17.82
21	8.57	15.97	8.75	25.07	17.99
22	8.37	14.75	8.53	22.00	16.63

TABLE 7
6x Fold Coverage: File Size (rounded to closest MB) and Compression Ratio for Chromosomes 11 and 20

	Raw	Illumina	QVZ	LEON	Dynamic bin
Chromosome 11					
SAM	3324	3324	3324	3324	3324
BAM	705	477	664	351	421
CRAM	399	214	357	135	174
Compression	8.33	15.51	9.30	24.60	19.08
Chromosome 20					
SAM	1409	1409	1409	1409	1409
BAM	300	205	298	154	190
CRAM	165	89	162	58	79
Compression	8.55	15.76	8.72	24.51	17.82

For CRAM, the compression ratio is always significantly better for Illumina, LEON and dynamic binning; for QVZ, the compression ratio is better but not significantly so.

5.2 Variant Calling Performance

Table 8 shows sensitivity, precision and F-score for variant calling performance of two different callers (GATK and Samtools), considering Illumina ground truth, for chromosomes 11 and 20.

TABLE 8
6x Fold Coverage: Variant (SNP) Calling Performance with Illumina Ground Truth for Chromosomes 11 and 20

	GATK			Samtools		
	Sens.	Prec.	F-Score	Sens.	Prec.	F-Score
Chromosome 11						
Raw	0.7137	0.9857	0.8279	0.7388	0.9810	0.8429
Illumina	0.7124	0.9881	0.8279	0.7414	0.9806	0.8444
QVZ	0.7135	0.9814	0.8263	0.7340	0.9780	0.8386
LEON	0.7109	0.9859	0.8261	0.7511	0.9742	0.8482
Dyn. bin	0.7125	0.9860	0.8273	0.7354	0.9811	0.8406
Chromosome 20						
Raw	0.6687	0.9805	0.7952	0.6963	0.9710	0.8110
Illumina	0.6662	0.9827	0.7941	0.7010	0.9710	0.8142
QVZ	0.6687	0.9755	0.7935	0.6918	0.9680	0.8069
LEON	0.6657	0.9804	0.7930	0.7038	0.9659	0.8142
Dyn. bin	0.6673	0.9808	0.7942	0.6930	0.9717	0.8090

TABLE 9
6x Fold Coverage: Variant (SNP) Calling Performance with NIST-GIAB Ground Truth for Chromosomes 11 and 20

	GATK			Samtools		
	Sens.	Prec.	F-Score	Sens.	Prec.	F-Score
Chromosome 11						
Raw	0.7624	0.7523	0.7573	0.7866	0.7461	0.7658
Illumina	0.7615	0.7546	0.7580	0.7891	0.7355	0.7667
QVZ	0.7618	0.7486	0.7551	0.7817	0.7441	0.7624
LEON	0.7595	0.7525	0.7559	0.7983	0.7397	0.7679
Dyn. bin	0.7612	0.7526	0.7569	0.7831	0.7464	0.7643
Chromosome 20						
Raw	0.7267	0.8464	0.7820	0.7534	0.8347	0.7920
Illumina	0.7243	0.8487	0.7816	0.7585	0.8345	0.7947
QVZ	0.7264	0.8418	0.7798	0.7490	0.8326	0.7886
LEON	0.7235	0.8466	0.7802	0.7602	0.8289	0.7931
Dyn. bin	0.7251	0.8466	0.7811	0.7501	0.8356	0.7906

Table 9 shows sensitivity, precision and F-score for variant calling performance of GATK and Samtools callers, considering the NIST-GIAB ground truth, again for chromosomes 11 and 20.

These tables show that for both callers and both ground truths, F-score with raw quality values differs very little from those of all lossy models, with some actually seeing a slight improvement in F-score. This is also the case for all 22 chromosomes; see full details in [34].

5.3 ROC Curve Analysis

The ROC curves for each of the lossy techniques and for raw quality scores, using Samtools caller and Illumina ground truth, are available in [34]. Similar to the 5x coverage experiments, they show that Illumina binning outperforms raw quality values; LEON is also visibly above at critical points. In general variant calling performance is not negatively affected and in some cases even improved.

The values for area under the curve (AUC) for these plots are presented in Table 10. In this table it is more evident that

TABLE 10
AUC, 6x fold coverage

Chr.	Caller	Raw	Illumina	QVZ	LEON	Dyn. bin
11	GATK	0.6990	0.7492	0.6991	0.7120	0.6923
11	Samtools	0.8257	0.8320	0.8262	0.8273	0.8200
20	GATK	0.6880	0.7195	0.6890	0.7005	0.6857
20	Samtools	0.7988	0.8068	0.7991	0.8028	0.7950

TABLE 11
High (50x) Coverage: File Size (rounded to closest MB) and Compression Ratio

	Raw	Illumina	QVZ	LEON	Dynamic bin
Chromosome 11					
SAM	44741	44741	44741	44741	44741
BAM	10669	8830	10531	7883	8680
CRAM	6685	5297	6527	4726	5191
Compression	6.69	8.44	6.85	9.46	8.61
Chromosome 20					
SAM	19978	19978	19978	19978	19978
BAM	4763	3952	4703	3530	3881
CRAM	2983	2370	2913	2116	2321
Compression	6.69	8.42	6.85	9.44	8.60

all values are close, and furthermore that Illumina, LEON and QVZ slightly outperform the case with the raw quality values.

In comparison with the 5x fold coverage experiments, observe that with this data set (6x) we obtain better accuracy for variant calling. This is not surprising because with higher coverage, we are more likely to have the correct information about each base and therefore to find the variants.

6 Results and Analysis – 50x Coverage

As in the previous experiments, the chromosomes analyzed are from individual NA12878. The alignment performed with BWA is provided by 1000 Genomes project [1] and the coverage is 50x. The BAM file for this data set is 254GB, therefore due to storage constraints all experiments were performed on chromosomes 11 and 20 only. As seen in the experiment performed with the whole genome with 6x coverage, the behaviour in all other chromosomes is expected to be similar in terms of compression ratio. The SAM files are 44.7GB for chromosome 11 and 20.0GB for chromosome 20.

6.1 Compression Ratio

Table 11 summarizes the sizes of the files as well as the compression ratio. In both chromosomes the compression ratio results are similar. LEON obtains the best compression ratio, followed by dynamic binning and Illumina binning, with QVZ last. Although we obtain an improvement in compression for all of the lossy models studied, note that in this experiment, the compression ratio is lower than that obtained with lower coverage. Roughly, the compression ratio for high coverage is half that obtained with 6x coverage and this property holds for the file with raw quality values as well as for the files for the lossy models. Also note that for each technique, as well as for the raw file, the pair of compression ratio values belonging to each chromosome are very similar and we would expect this in every chromosome.

6.2 Variant Calling Performance

To analyze the impact of adjusted quality values on the variant calling process see Table 12, which summarizes sensitivity, precision and F-score for each lossy model as well as for the original (raw) quality values. Scores are reported for each caller, GATK and Samtools. For the results in this

TABLE 12
Variant (SNP) calling performance with Illumina ground truth, chromosome 11 (50x fold coverage)

	GATK			Samtools		
	Sens.	Prec.	F-Score	Sens.	Prec.	F-Score
Chromosome 11						
Raw	0.9778	0.9669	0.9723	0.9785	0.9590	0.9686
Illumina	0.9752	0.9718	0.9735	0.9791	0.9588	0.9688
QVZ	0.9778	0.9668	0.9723	0.9781	0.9592	0.9685
LEON	0.9776	0.9634	0.9704	0.9815	0.9512	0.9661
Dyn. bin	0.9777	0.9668	0.9722	0.9783	0.9580	0.9680
Chromosome 20						
Raw	0.9549	0.9586	0.9568	0.9577	0.9469	0.9523
Illumina	0.9513	0.9653	0.9582	0.9586	0.9469	0.9527
QVZ	0.9550	0.9587	0.9569	0.9581	0.9465	0.9522
LEON	0.9547	0.9549	0.9548	0.9622	0.9369	0.9494
Dyn. bin	0.9548	0.9586	0.9567	0.9579	0.9459	0.9518

table we considered Illumina ground truth. For details on the same experiments with GIAB-NIST ground truth, and also the actual numbers of true positives, false positives and false negatives, see [34]. Again, for both ground truths, both callers, and both chromosomes the relative results of the different lossy models are very similar.

The F-scores provide evidence that in general the variant calling performance is not affected by any of the lossy techniques. There are two facts to highlight. Firstly, according to the F-score measure, the variant calling performance is improved when applying Illumina binning to the quality scores. Secondly, when the LEON model is applied to the quality values, the F-score obtained is the lowest one, for both chromosomes and both callers. It is important to mention this because the LEON model reports a considerably better compression ratio than the other techniques. We can see this as evidence of the trade-off between compression ratio and variant calling performance.

6.3 ROC Curve Analysis

Consider Figure 3, the ROC curve for Illumina ground truth and Samtools caller for chromosome 11. Note that with high coverage the variant calling performance is much better than that with lower coverage: for false positive rate around 0.2, true positive rate is already above 0.8.

As for comparing the curves for each of the lossy techniques versus that for the original quality values, observe that before all curves converge the curve for LEON (green) is below the one for raw quality values (red). Also, the curve for Illumina (blue) is above the one for the raw quality values, confirming the slight improvement previously noted in the F-score for Illumina binning. The curves for both QVZ and dynamic binning mostly overlap that for raw quality values. We consider this a good result because it implies that variant calling performance is not negatively affected even though both achieved a better compression ratio than that for the original quality scores.

To summarize the information from ROC curves, we consider the AUC as another metric for the variant calling performance. Table 13 reports AUC values for both callers, GATK and Samtools. By comparing the AUC for each technique we notice that there is no indication that lossy models negatively affect the variant calling performance as all of them are either greater than the AUC with raw quality values or very close.

TABLE 13
AUC, 50x fold coverage

Chr.	Caller	Raw	Illumina	QVZ	LEON	Dyn. bin
11	GATK	0.8205	0.8439	0.8192	0.8250	0.8191
11	Samtools	0.8792	0.8804	0.8797	0.8713	0.8797
20	GATK	0.7858	0.8129	0.7878	0.7946	0.7865
20	Samtools	0.8952	0.8988	0.8954	0.8926	0.8954

7 Conclusions and Future Work

We introduced a new lossy technique for quality scores, dynamic binning, and studied it along with three other techniques: Illumina binning, QVZ, and LEON. In particular we analyzed the effect of each of these on the CRAM compression format. We examined not only the compression ratio, but also the effect on variant calling performance.

According to the three experiments presented, all using different coverage data sets, the four lossy techniques all improve the compression ratio. Nonetheless, it is worthwhile noting that the compression ratio is lower when coverage is higher. As coverage increases, reads are still easily compressed as there is a great deal of redundancy; however, this is not the case for the quality scores. As a result, less compression is possible. However, more information also implies better variant calling performance, as confirmed by F-score and AUC. For example, for chromosome 20 the compression ratio for raw quality values varies from 13.40 (5x coverage) to 8.55 (6x coverage) to 6.69 (50x coverage). And for each lossy model, the compression ratio varies similarly. Meanwhile, the corresponding F-score for Samtools caller varies from 0.7762 (5x coverage) to 0.8110 (6x coverage) to 0.9523 (50x coverage).

It is also worthwhile noting that the lossy compression methods improve the compression ratio of the CRAM files by a greater percentage than they improve the compression ratio of the BAM files. Again, this effect is reduced as the coverage increases. However, since the CRAM format is already a significant compression compared to BAM, this provides further motivation for using the CRAM format.

In terms of compression ratio, LEON gives the best results, more than doubling the compression ratio obtained with raw quality values for 5x coverage and 6x coverage. Dynamic binning also doubles or nearly doubles the compression ratio obtained by the raw quality values for these coverages. As for 50x coverage, all lossy techniques improve upon the compression ratio obtained by the raw quality values, but not as markedly. For example, LEON has the best compression ratio but improves from only 6.69 to 9.44 for chromosome 11 and 6.69 to 9.46 for chromosome 20.

However, with respect to variant calling performance, LEON reports a lower F-score than all other techniques, across all coverages, chromosomes and callers considered, with the sole exception of 6x coverage and Samtools caller.

Both dynamic binning and Illumina binning improved the compression ratio without compromising variant calling performance. In general dynamic binning obtained a better compression ratio than Illumina binning. Although

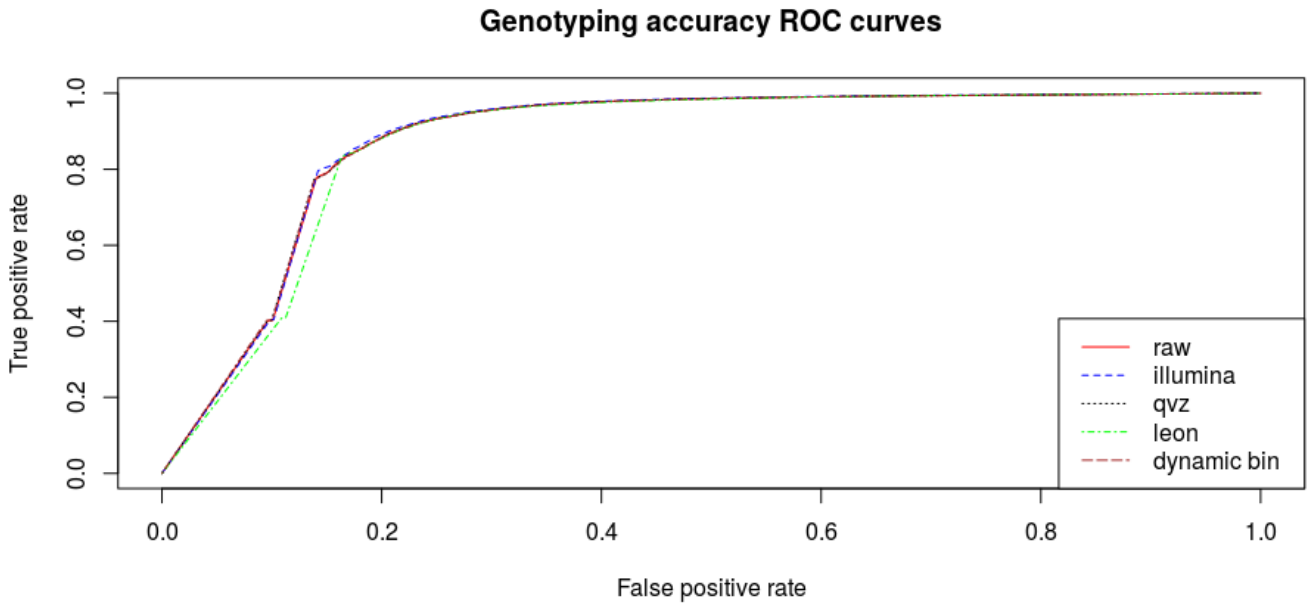

Fig. 3. ROC, chromosome 11 (50x fold coverage). True positive and false positive rates are as defined in Section 3.4 and relate to SNP calling only.

with dynamic binning the improvement of variant calling performance did occur in some cases, Illumina binning was more consistent in boosting variant calling performance as measured by F-score.

In all experiments QVZ showed the lowest compression ratio of all and was only slightly above the compression ratio obtained with the raw quality values, although the variant calling performance remained very close.

In comparing the techniques, there is an interplay between several different aspects. It is important to consider how much compression may help in different situations and also the effect of different parameters for each of the techniques. One important aspect is the actual datasets being examined: as noted above, as coverage increases there is typically lower compression possible in the quality values (due to relatively lower redundancy in comparison to the reads) but also some datasets contain other tags that take up significant space. As another example, QVZ may be at a disadvantage in comparison to the other methods for CRAM compression as it may still have a high number of quality scores present. Also, due to the nature of CRAM compression, the frequency with which values are repeated has a direct effect on the level of compression achievable. Dynamic binning uses 5 bins, so in general the decreased alphabet is easier to compress, however with larger files there are more blocks and so more opportunities for bins in different blocks to have different representatives, making compression harder. Notice that the difference in relative compression ratios between the different methods decreases as coverage increases. On a related note, depending on the sequencing technology used, the number of quality values used may vary significantly which in turn impacts the compression achievable. For example, Illumina's newer technology [18] uses a greatly reduced number of quality scores. In short, the type of data has a definite impact on the relative success of the various techniques. All of the current techniques for compression of quality values, including those discussed in this study, may help to provide insight on how to assign scores to be used in newer technologies.

TABLE 14
F-score comparison for no quality values. Chromosome 20, ground truth Illumina and Samtools caller.

	5x	6x	50x
Raw	0.7762	0.8237	0.9523
No quality values	0.7180	0.7682	0.8796

This study has shown that by adjusting quality values the compression ratio can be improved without compromising SNP calling performance. Consider Table 14, which gives the F-scores achieved for constant quality values and for raw quality values: we see that when setting all quality values to a constant value the F-score drops significantly, indicating that the information provided by the quality scores is still necessary for good SNP calling performance.

Nowadays many approaches for compression of next-generation DNA sequencing data output are being studied. Lossy techniques can be very useful in this area, as it has been shown that variant calling performance is the same or sometimes even better when adjusting the quality values.

There remain several studies to continue improving these techniques. The noise that quality values present in their distribution needs to be well understood by making further analysis of quality values behaviour and statistics.

As future work, it would be useful to consider other types of mutations such as indels. It would also be instructive to examine the performance of the various techniques on different datasets. One such newer dataset is Syndip [23]. As with the datasets used in this project [30][40], Syndip was constructed via a consensus of multiple callers to reduce

bias; also, its developers aimed to avoid possible additional bias that may be present in earlier datasets due to the same algorithms being used for both construction and testing.

Instead of a single standard lossy technique, it may be preferable to develop several options which the user can select depending on the project, as we observed different behaviours in our experiments by varying only fold coverage. Many other factors can vary including for example sequencing technology, which as noted above can have a significant impact. As new methods and technologies are developed, further comparisons will need to be performed to evaluate which methods are best in which circumstances, and to help develop new ideas.

For the exploratory idea we developed with dynamic binning, there are several other paths to continue studying. This includes further consideration of the number of bins, the ranges covered by each bin, and the lengths of the blocks. The reduced number of bins used in Illumina's newer technology [18] should be taken into account while addressing these considerations, although as noted in Section 3.5 we also see that dynamic binning allows for more accurate quality values and therefore is one means of helping to inform decisions about binning in the various technologies. We note also the similarity of dynamic binning to P-block and R-block [7], which are also flexible but which use variable blocks. Although in the current definition of dynamic binning all blocks are fixed, such flexibility could also be incorporated into dynamic binning, along with consideration of various probabilities.

ACKNOWLEDGMENTS

This research was funded in part by the Natural Sciences and Engineering Research Council of Canada. The authors would like to thank Dr. Ping Liang for helpful discussions, as well as the anonymous reviewers for their very useful comments and suggestions.

REFERENCES

[1] 1000 Genomes Project Repository. ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/phase3/data /na12878. Accessed: 2016-06-15.
[2] H. Afify, M. Islam, and M.A. Wahed. Dna lossless differential compression algorithm based on similarity of genomic sequence database. *International Journal of Computer Science & Information Technology (IJCSIT)*, 3(4):145–154, August 2011.
[3] G. Benoit et al. Reference-free compression of high throughput sequencing data with a probabilistic de bruijn graph. *BMC Bioinformatics*, 16:288, April 2015.
[4] V. Bhola et al. No-reference compression of genomic data stored in fastq format. In *Bioinformatics and Biomedicine (BIBM), 2011 IEEE International Conference on*, pages 147–150. IEEE, 2011.
[5] James K Bonfield, Shane A McCarthy, and Richard Durbin. Crumble: reference free lossy compression of sequence quality values. *Bioinformatics*, 35(2):337–339, 2018.
[6] J.K. Bonfield and M.V. Mahoney. Robust relative compression of genomes with random access. *PLoS One*, 8:e59190, March 2013.
[7] R. Cánovas, A. Moffat, and A. Turpin. Lossy compression of quality scores in genomic data. *Bioinformatics*, 30(15):2130–2136, August 2014.
[8] P.J.A. Cock et al. The sanger fastq file format for sequences with quality scores, and the solexa/illumina fastq variants. *Nucleic Acids Research*, 38(6):1767–1771, April 2010.
[9] CRAM Format Specification. https://samtools.github.io/hts-specs/cramv3.pdf. Accessed: 2019-05-12.
[10] CRAM Format Specification. https://www.ga4gh.org/news/cram-compression-for-genomics/.
[11] S. Deorowicz and S. Grabowski. Robust relative compression of genomes with random access. *Bioinformatics*, 27(21):2979–2986, September 2011.
[12] M. DePristo et al. A framework for variation discovery and genotyping using next-generation dna sequencing data. *NATURE GENETICS*, 43:491–498, 2011.
[13] S. Grabowski, S. Deorowicz, and L. Roguski. Disk-based compression of data from genome sequencing. *Bioinformatics*, 31:844, December 2014.
[14] GRCH37 Reference Genome. http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/reference. Accessed: 2016-06-15.
[15] S. Grumbach and F. Tahi. Compression of dna sequences (extended abstract). 1994.
[16] F. Hach et al. Scalce: boosting sequence compression algorithms using locally consistent encoding. *Bioinformatics*, 28(23):3051–3057, December 2012.
[17] F. M. Hsi-Yang et al. Efficient storage of high throughput dna sequencing data using reference-based compression. *Genome Research*, 21(5):734–740, May 2011.
[18] Illumina: Analyze Data. Novaseq 6000 system quality scores and rta3 software.
[19] Illumina White Paper: Informatics. Reducing whole-genome data storage footprint.
[20] L. Janin, O. Schulz-Trieglaff, and A.J. Cox. Beetl-fastq: a searchable compressed archive for dna reads. *Bioinformatics*, 30(19):2796–2801, October 2014.
[21] H. Li and R. Durbin. Fast and accurate short read alignment with burrows-wheeler transform. *Bioinformatics*, 25:1754–60, 2009.
[22] H. Li et al. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–9, 2009.
[23] Heng Li, Jonathan M Bloom, Yossi Farjoun, Mark Fleharty, Laura Gauthier, Benjamin Neale, and Daniel MacArthur. A synthetic-diploid benchmark for accurate variant-calling evaluation. *Nature methods*, 15(8):595, 2018.
[24] G. Malysa et al. Qvz: lossy compression of quality values. *Bioinformatics*, 31(19):3122–3129, October 2015.
[25] A. McKenna et al. The genome analysis toolkit: a mapreduce framework for analyzing next-generation dna sequencing data. *GENOME RESEARCH*, 20:1297–1303, 2010.
[26] I. Ochoa et al. Qualcomp: a new lossy compressor for quality scores based on rate distortion theory. *BMC Bioinformatics*, 14:187, June 2013.
[27] I. Ochoa et al. Effect of lossy compression of quality scores on variant calling. *Brief Bioinform*, March 2016.
[28] R. Patro and C. Kingsford. Data-dependent bucketing improves reference-free compression of sequencing reads. *Bioinformatics*, 32(14):248, April 2015.
[29] Picard Tools. http://broadinstitute.github.io/picard. Accessed: 2016-06-15.
[30] Platinum Genomes. http://www.illumina.com/platinumgenomes. Accessed: 2016-06-15.
[31] SAM Format Specification. http://samtools.github.io/hts-specs/samv1.pdf. Accessed: 2016-06-15.
[32] K. Sayood. *Introduction to Data Compression, Third Edition*. Morgan Kaufmann Publishers Inc., 2005.
[33] Z.D. Stephens et al. Big data: Astronomical or genomical? *PLoS Biol*, 13(7):e1002195, 2015.
[34] V. Suaste Morales. Lossy compression of quality values in next-generation sequencing data (http://hdl.handle.net/10464/11527). Master's thesis, Brock University, 2017.
[35] G.A. Van der Auwera et al. From fastq data to high-confidence variant calls: The genome analysis toolkit best practices pipeline. *Current protocols in bioinformatics*, 43:11.10.1–11.10.33, 2013.
[36] VCF Format Specification. http://samtools.github.io/hts-specs/vcfv4.2.pdf. Accessed: 2016-06-15.
[37] Jan Voges, Jörn Ostermann, and Mikel Hernaez. Calq: compression of quality values of aligned sequencing data. *Bioinformatics*, 34(10):1650–1658, 2017.
[38] Y.W Yu et al. Traversing the k-mer landscape of ngs read datasets for quality score sparsification. *Research in Computational Molecular Biology. RECOMB 2014. Lectures Notes in Computer Science*, 8394:385–399, 2014.
[39] Y.W Yu et al. Quality score compression improves genotyping accuracy. *Nature Biotechnology*, pages 240–243, 2015.
[40] J.M. Zook et al. Integrating human sequence data sets provides a resource of benchmark snp and indel genotype calls. *Nature Biotechnology*, pages 246–251, February 2014.