A cluster-based approach to compression of Quality Scores

Mikel Hernaez, Idoia Ochoa, Tsachy Weissman

Department of Electrical Engineering, Stanford University
{mhernaez,iochoa,tsachy}@stanford.edu

Abstract

Massive amounts of sequencing data are being generated thanks to advances in sequencing technology and a dramatic drop in the sequencing cost. Storing and sharing this large data has become a major bottleneck in the discovery and analysis of genetic variants that are used for medical inference. As such, lossless compression of this data has been proposed. Of the compressed data, more than 70% correspond to quality scores, which indicate the sequencing machine reliability when calling a particular basepair. Thus, to further improve the compression performance, lossy compression of quality scores is emerging as the natural candidate. Since the data is used for genetic variants discovery, lossy compressors for quality scores are analyzed in terms of their rate-distortion performance, as well as their effect on the variant callers. Previously proposed algorithms do not do well under all performance metrics, and are hence unsuitable for certain applications.

In this work we propose a new lossy compressor that first performs a clustering step, by assuming all the quality scores sequences come from a mixture of Markov models. Then, it performs quantization of the quality scores based on the Markov models. Each quantizer targets a specific distortion to optimize for the overall rate-distortion performance. Finally, the quantized values are compressed by an entropy encoder. We demonstrate that the proposed lossy compressor outperforms the previously proposed methods under all analyzed distortion metrics. This suggests that the effect that the proposed algorithm will have on any downstream application will likely be less noticeable than that of previously proposed lossy compressors. Moreover, we analyze how the proposed lossy compressor affects Single Nucleotide Polymorphism (SNP) calling, and show that the variability introduced on the calls is considerably smaller than the variability that exists between different methodologies for SNP calling.[1]

1 Introduction

Recent advancements in Next Generation high-throughput Sequencing (NGS) have led to a drastic reduction in the cost of sequencing a genome (http://goo.gl/kKvmDl). This has generated an unprecedented amount of genomic data that must be stored, processed, and transmitted. To facilitate this effort, data compression techniques that allow for more efficient storage as well as fast exchange and dissemination of these data have been proposed in the literature.

The raw genomic data is mainly comprised of the reads (fragments of the genome) and the sequence of quality scores[2]. The quality scores are generally stored using the *Phred* scale, which corresponds to the particular number $Q = \lceil -10 \log_{10} P \rceil$, where P is an estimate (calculated by the base calling software running on the sequencing

[1] The code used in this work is available at *https://github.com/mikelhernaez/qvz2*

[2] Also referred to as quality values.

machine) of the probability that the corresponding nucleotide in the read is in error. These scores are commonly represented with the ASCII alphabet [33 : 73], where the value corresponds to Q + 33.

When losslessly compressed, quality scores comprise more than 70% of the compressed file [1]. In addition, it has been shown that the quality scores are inherently noisy [2], and downstream applications that use them do so in varaying heuristic manners. For these reasons, lossy compression of quality scores has been proposed to further reduce the storage requirements at the cost of introducing a distortion (i.e., the reconstructed quality scores may differ from the original ones).

The data under consideration is used for biological inference, and thus it is important to analyze how lossy compression affects this inference. Since different downstream applications exist that use the data for different purposes, it is not feasible to analyze the effect in all of them. As a result, the effort has been focused on analyzing the effect on SNP calling, as it is one of the most widely used downstream applications in practice. Additionally, it is standard practice to perform a rate-distortion analysis independent of downstream applications, but from which insight can be gained into the effect that a lossy compressor will have on a downstream application.

Among the several lossy compressors proposed in the recent literature (see [3, 4, 5] and references therein), none excel under in all metrics. In this work we propose a new lossy compressor for quality scores and show that it improves upon the previously proposed lossy compressors for quality scores in rate-distortion. Moreover, this improvement is consistent across all chosen distortion criteria, in contrast to previously proposed methods that perform poorly under a subset of distortion criteria [3, 4]. We further analyze the effect that the proposed lossy compressor has on SNP calling, and show that the variability introduced in the calls is smaller than the variability observed between the most common SNP callers used in practice. This suggests that lossy compression could be used to boost compression performance without compromising the discovery of genetic variants.

The proposed lossy compressor for quality scores performs a clustering step prior to compression. The clustering method is based on assuming that the set of quality score sequences come from a Markov mixture model. After the clustering step, the algorithm quantizes the quality scores based on the Markov models. The distortion at each quantizer is chosen to maximize the overall rate-distortion performance. Finally, the quantized values are compressed with an adaptive arithmetic encoder. Next we describe the proposed method in detail.

2 Clustering based on Markov Mixture Model

We denote by $\mathcal{Q} = \{Q_i\}_{i=1}^N$ the set of all quality value sequences found in a genomic data file. For simplicity, we assume that all the sequences are of the same length T, thus, $Q_i = \{Q_{i,t}\}_{t=1}^T$. Without loss of generality we assume $Q_{i,t} \in \mathcal{A} = \{1, \ldots, |\mathcal{A}|\}$.

Let us first consider the case where the sequences are generated by an order-1 Markov source. That is, the probability of each sequence Q_i is given by

$$P(Q_i) = \prod_{t=1}^{T} P(Q_{i,t}|Q_{i,t-1}, \ldots, Q_{i,1}) = P(Q_{i,1}) \prod_{t=2}^{T} P(Q_{i,t}|Q_{i,t-1}), \qquad (1)$$

where the last equality comes from the Markov assumption.

A discrete Markov source can be fully determined by its transition matrix $\mathbf{A}$, where $A_{mn} = P(Q_{i,t} = m|Q_{i,t-1} = n)$ is the probability of going from state n to state m, $\forall t$, and the prior state probability $\pi_n = P(Q_{i,1} = n)$, which is the probability of starting at state n. We further denote the model parameters as $\theta = \{\mathbf{A}, \pi\}$.With this notation we can rewrite (1) as

$$P(Q_i; \theta) = \prod_{n=1}^{|\mathcal{A}|} (\pi_n)^{\mathbb{1}[Q_{i,1}=n]} \prod_{t=2}^{T} \prod_{m=1}^{|\mathcal{A}|} \prod_{n=1}^{|\mathcal{A}|} (A_{mn})^{\mathbb{1}[Q_{i,t}=m, Q_{i,t-1}=n]}. \tag{2}$$

The States Space of the Markov Models

Previously, we have assumed that the stochastic process that generates the quality value sequences is time invariant, i.e., that the value of A_{mn} is independent of the time t. However, strong correlations exist between adjacent quality scores, as well as a trend that the quality scores degrade as a read progresses.

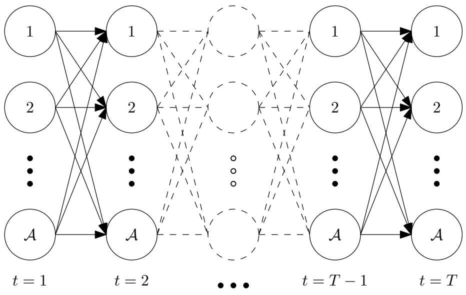

Figure 1: Our temporal Markov model.

In order to take into consideration the temporal behavior of the quality value sequences, we increase the number of states from $\mathcal{A}$ to $|\mathcal{A}| \times T$, one for each possible value of Q and t. Fig. 1 shows the diagram of the state space and the allowed transitions between states. To represent the temporal dimension, we redefine the transition matrix as a three dimensional matrix, where the first dimension represents the previous value of the quality score, the second one the current value of the quality score and the third one the time t within the sequence. That is, $A_{mnt} = P(Q_{i,t} = m|Q_{i,t-1} = n)$ is the probability of transitioning from state n to state m at time t.

Markov Mixture Models

In this work we further assume that the quality value sequences have been generated independently by one of K underlying Markov models, such that the whole set of quality value sequences are generated by a mixture of Markov models. With some abuse of notation we now define $\theta = \{\pi^{(k)}, \mathbf{A}^{(k)}\}_{k=1}^{K}$ to be the parameters of the K Markov models, and $\theta_k = \{\pi^{(k)}, \mathbf{A}^{(k)}\}$ to be the parameters of the kth Markov model. We further define Z_i to be the latent random variable that specifies the identity of the mixture component for the ith sequence. Thus, the set of quality value sequences that has been generated by the sequencing machine is distributed as:

$$\mathcal{Q} \sim P(\mathcal{Q}; \theta) = \prod_{i=1}^{N} P(Q_i; \theta) = \prod_{i=1}^{N} \sum_{k=1}^{K} P(Q_i|Z_i = k; \theta)\mu_k, \tag{3}$$

where $\mu_k \triangleq P(Z_i = k)$, and $P(Q_i|Z_i = k; \theta)$ is the probability that the sequence Q_i has been generated by the kth Markov model. Substituting (2) in (3) we get that the likelihood of the data is given by

$$P(\mathcal{Q};\theta)=\prod_{i=1}^{N}\sum_{k=1}^{K}\mu_k\left(\prod_{n=1}^{|\mathcal{A}|}(\pi_n^{(k)})^{\mathbb{1}[Q_{i,1}=n]}\prod_{t=2}^{T}\prod_{m=1}^{|\mathcal{A}|}\prod_{n=1}^{|\mathcal{A}|}(A_{mnt}^{(k)})^{\mathbb{1}[Q_{i,t}=m,Q_{i,t-1}=n]}\right). \tag{4}$$

The goal of the clustering step is to assign each sequence to the most probable model that has generated it. However, since the parameters of the models are unknown, the clustering step first computes the maximum likelihood estimation of the parameters $\{\mathbf{A}^{(k)},\pi^{(k)},\mu_k\}$ of each of the Markov models. Since the log likelihood $\ell(\theta)\triangleq\log P(\mathcal{Q};\theta)$ is intractable due to the summand appearing in (4), this operation is done by using the well known Expectation-Maximitation (EM) algorithm [6]. The EM algorithm iteratively maximizes the function

$$g(\theta,\theta^{(l-1)})\triangleq\mathbb{E}_{Z|Q,\theta^{(l-1)}}\left[\sum_i\log P(Q_i,Z_i;\theta)\right],$$

which is the expectation of the complete log likelihood with respect to the conditional distribution of Z given Q and the current estimated parameters. It can be shown [6] that for any mixture model this function is given by

$$g(\theta,\theta^{(l-1)})=\sum_i\sum_k r_{ik}\log\mu_k+\sum_i\sum_k r_{ik}\log P(Q_i;\theta_k), \tag{5}$$

where $r_{ik}\triangleq P(Z_i=k|Q_i,\theta^{(l-1)})$ is the responsibility that cluster k takes for the quality sequence i. In particular, for the case of a mixture of Markov models, the previous equation is given by

$$\begin{aligned}g(\theta,\theta^{(l-1)})=\sum_{i=1}^{N}\sum_{k=1}^{K}r_{ik}\log\mu_k+\sum_{i=1}^{N}\sum_{k=1}^{K}r_{ik}\sum_{n=1}^{|\mathcal{A}|}\mathbb{1}[Q_{i,1}=n]\log(\pi_n^{(k)})+&\\ \sum_{i=1}^{N}\sum_{k=1}^{K}r_{ik}\sum_{t=2}^{T}\sum_{m=1}^{|\mathcal{A}|}\sum_{n=1}^{|\mathcal{A}|}\log(A_{mnt}^{(k)})\mathbb{1}[Q_{i,t}=m,Q_{i,t-1}=n],&\end{aligned} \tag{6}$$

where the expansion of $\log P(Q_i;\theta_k)$ is obtained by taking the log of (2).

The initialization of the EM algorithm is performed by randomly selecting the parameters $\{\widehat{\mathbf{A}}^{(k)},\hat{\pi}^{(k)},\hat{\mu}_k\}$. Then, the algorithm iteratively performs as follows. In the E-step it computes r_{ik}, which for the case of Markov mixture models is given by

$$r_{ik}\propto\hat{\mu}_k\hat{\pi}^{(k)}(Q_{i,1})\widehat{A}_2^{(k)}(Q_{i,1},Q_{i,2})\widehat{A}_3^{(k)}(Q_{i,2},Q_{i,3})\dots\widehat{A}_T^{(k)}(Q_{i,T-1},Q_{i,T}), \tag{7}$$

where

$$\widehat{A}_t^{(k)}(Q_{i,t-1},Q_{i,t})=\prod_{m=1}^{|\mathcal{A}|}\prod_{n=1}^{|\mathcal{A}|}(\widehat{A}_{mnt}^{(k)})^{\mathbb{1}[Q_{i,t}=m,Q_{i,t-1}=n]}.$$

In the M-step it computes the parameters $\hat{\theta}$ that maximize $Q(\theta,\theta^{(l-1)})$. In the case of a mixture of Markov models, these parameters can be computed using the Lagrange multipliers method on $Q(\theta,\theta^{(l-1)})$, where the constrains are that all the rows of $\mathbf{A}^{(k)}$

and the vectors $\pi^{(k)}$ and μ must sum to one. For the case under consideration, it can be shown that the maximizing parameters computed in the M-step are given by

$$\begin{aligned}\hat{\mu}_k &= \frac{1}{N}\sum_{i=1}^{N} r_{ik} && (8)\\ \hat{\pi}_n^{(k)} &= \frac{\sum_{i=1}^{N} r_{ik}\mathbb{1}[Q_{i,1}=n]}{\sum_{n'=1}^{|\mathcal{A}|}\sum_{i=1}^{N} r_{ik}\mathbb{1}[Q_{i,1}=n']} && (9)\\ \widehat{A}_{mnt}^{(k)} &= \frac{\sum_{i=1}^{N} r_{ik}\mathbb{1}[Q_{i,t}=m, Q_{i,t-1}=n]}{\sum_{m'=1}^{|\mathcal{A}|}\sum_{i=1}^{N} r_{ik}\mathbb{1}[Q_{i,t}=m', Q_{i,t-1}=n]}, && (10)\end{aligned}$$

with $t = 2, \ldots, T$, $n = 1, \ldots, |\mathcal{A}|$ and $m = 1, \ldots, |\mathcal{A}|$.

Furthermore, the EM algorithm guarantees that choosing θ to improve $Q(\theta, \theta^{(l-1)})$ beyond $Q(\theta^{(l-1)}, \theta^{(l-1)})$ will improve $\ell(\theta)$ beyond $\ell(\theta^{(l-1)})$, which yields a decreasing value of $Q(\theta^{(l-1)}, \theta^{(l-1)})$ per iteration. We stop the algorithm once the change on the value of $Q(\theta^{(l-1)}|\theta^{(l-1)})$ is small enough, or after a fixed number of iterations.

Once the EM algorithm terminates, the value of r_{ik} tells us the responsibility of each mixture component k over the sequence i. The clustering step uses this information to perform the clustering. Specifically, each sequence is assigned to the cluster $\mathcal{C}_k$ with k such that $r_{ik} \geq r_{ik'}$, $\forall k' \neq k$.

3 Quantization Step

As described previously, we have modeled the data using a mixture of Markov models. This mixture has generated an underlying probability model that will be used to design a codebook for the compression of the quality scores. The codebook is a set of quantizers indexed by the cluster id k, the position t within the read and the previously quantized value (the context). These quantizers are constructed using a tailored version of the discrete Lloyd's algorithm [7]. After quantization, a lossless, adaptive arithmetic encoder is applied to achieve entropy-rate compression. Next we describe the quantizer in detail.

Given a random variable X governed by the probability mass function $P(\cdot)$ over the alphabet $\mathcal{X}$ of size K, let $\mathbf{D} \in \mathbb{R}^{K\times K}$ be a distortion matrix where each entry $D_{x,y} = d(x, y)$ is the penalty for reconstructing symbol x as y. We further define $\mathcal{Y} \subseteq \mathcal{X}$ to be the alphabet of the quantized values of size $M \leq K$.

The quantizer, denoted hereafter as $LM(\cdot)$, is a mapping $\mathcal{X} \to \mathcal{Y}$ that minimizes the expected distortion. Specifically, the quantizer seeks to find a collection of boundary points $b_k \in \mathcal{X}$ and reconstruction points $y_k \in \mathcal{Y}$, where $k \in \{1, 2, \ldots, M\}$, such that the quantized value of symbols $x \in \mathcal{X}$ is given by the reconstruction point of the region to which it belongs. That is, the quantizer aims to minimize

$$\{b_k, y_k\}_{k=1}^{M} = \underset{b_k, y_k}{\operatorname{argmin}} \quad \sum_{j=1}^{M} \sum_{x=b_{j-1}}^{b_j - 1} P(x) d(x, y_j). \tag{11}$$

In order to solve this problem we perform a one-dimensional weighted k-means algorithm, where after initializing the boundary points b_k, the algorithm iteratively performs as follows: i) for each region k choose the $y_k \in \{b_{k-1}, \ldots, b_k - 1\}$ that minimizes $\sum_{x=b_{k-1}}^{b_k-1} P(x)d(x,y)$, and ii) assign each point x to the closest reconstructed point y_k, where the distance is measured as $d(x,y)$, yielding new boundary points b_k. The algorithm stops if no further change is obtained in the b_k or after a fixed number of iterations.

Given a distortion matrix $\mathbf{D}$, the defined quantizer depends on the number of regions M and the input probability mass function $P(\cdot)$. Thus we denote the quantizer with M regions as $LM_M^P(\cdot)$, and the quantized value of a symbol $x \in \mathcal{X}$ as $LM_M^P(x)$. Note that a reconstructed point y has probability of occurrence $P(y) = \sum_{x:LM_M^P(x)=y} P(x)$. Thus, each generated quantizer $LM_M^P(\cdot)$ defines a rate-distortion pair, where the rate and distortion are given by

$$R(LM_M^P(\cdot)) = \sum_{y\in\mathcal{Y}} P(y)\log_2 P(y) \quad \text{and} \quad D(LM_M^P(\cdot)) = \sum_{y\in\mathcal{Y}} \sum_{x:LM_M^P(x)=y} P(y)d(x,y),$$

respectively. Furthermore, for a fixed probability mass function $P(\cdot)$, the only varying parameter is the number of regions M. Since M needs to be an integer, not all rate-distortion pairs are achievable. Thus, as done in QVZ [3], we define an extended version of the LM quantizer, which consists of two LM quantizers with the number of regions given by ρ and $\rho+1$, each of them used with probability $1-r$ and r, respectively (where $0 \leq r \leq 1$). In contrast to QVZ, in this work we are interested in achieving an arbitrary distortion D; therefore, ρ is given by the maximum number of regions such that $D(LM_\rho^P(X)) > D$ (which implies $D(LM_{\rho+1}^P(X)) < D$). Then, the probability r is chosen such that the average distortion is equal to D.

The reason for setting all quantizers to the same distortion D is the following. Given that there are a maximum of quantizers $|\mathcal{A}| \times T \times K$ (indexed by previously quantized value, position and cluster id), the final rate R is given by the convex combination of the individual rates R_i of all the quantizers. Thus, one can pose the following optimization problem:

$$\begin{aligned} \underset{R_i}{\text{minimize}} \quad & \sum_i \alpha_i Ri \\ \text{subject to} \quad & \sum_i \alpha_i K_i \exp(-h_i R_i) = D, \end{aligned}$$

where we have assumed that the rate-distortion function generated by each of the quantizers is of the form $D_i(R_i) = K_i \exp(-h_i R_i)$ [8]. Solving this problem using the Lagrange multipliers method, we obtain that the optimal distortion at which each quantizer must operate is given by

$$D_i = \frac{D}{h_i \sum_i \frac{\alpha_i}{h_i}}.$$

For the case under consideration, h_i may not be computable in some cases. Moreover, we expect all quantizers to exhibit a similar behavior. Thus, we assume $h_i = h \; \forall i$, which translates into all quantizers targeting the same distortion D.

Finally, due to the space constraint, we refer the reader to [3] for a detailed explanation of the computation of the quantizer input probability $P(\cdot)$ and the codebook generation.

Table 1: Distortion metrics used for assessment of the lossy compressors in terms of rate-distortion performance. Q is the original quality score and $\widehat{Q}$ is the reconstructed one after lossy compression.

MSE	$d_{mse}(Q,\widehat{Q}) = \frac{1}{L \cdot N}\sum_{i=1}^{N}\sum_{j=1}^{L}\lvert Q_{i,j}-\widehat{Q}_{i,j}\rvert^2$
L1	$d_{\ell 1}(Q,\widehat{Q}) = \frac{1}{L \cdot N}\sum_{i=1}^{N}\sum_{j=1}^{L}\lvert Q_{i,j}-\widehat{Q}_{i,j}\rvert$
Lorentzian	$d_L(Q,\widehat{Q}) = \frac{1}{L \cdot N}\sum_{i=1}^{N}\sum_{j=1}^{L}\log(1+\lvert Q_{i,j}-\widehat{Q}_{i,j}\rvert)$
Chebyshev	$d_C(Q,\widehat{Q}) = \frac{1}{N}\sum_{i=1}^{N}\max_{1\leq j\leq L}\lvert Q_{i,j}-\widehat{Q}_{i,j}\rvert)$
Max-Min	$d_{MM}(Q,\widehat{Q}) = \frac{1}{N}\sum_{i=1}^{N}\max_{1\leq j\leq L}\frac{\max(Q_{i,j},\widehat{Q}_{i,j})}{\min(Q_{i,j},\widehat{Q}_{i,j})}$
Soergel	$d_S(Q,\widehat{Q}) = \frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{j=1}^{L}\lvert Q_{i,j}-\widehat{Q}_{i,j}\rvert}{\sum_{j=1}^{L}\max(Q_{i,j},\widehat{Q}_{i,j})}$

4 Results

To assess the performance of the proposed algorithm, we use data from the individual NA12878. Specifically, we extracted the chromosome 20 of two Illumina pair-end, whole genome sequencing datasets, one with 15× coverage (composed of almost 9 million lines with 101 quality scores per line) and the other with 30× coverage (composed of more than 20 million lines with 101 quality scores per line).

We carry out two analyses. First, we compare the performance of the proposed algorithm to the state-of-the-art lossy compressors for quality scores in terms of rate-distortion performance, for several distortion metrics. The reason for performing the rate-distortion analysis over different metrics is that there is a wide variety of downstream applications that use the quality scores in widely varying heuristic manners. Thus, an algorithm that performs well in terms of rate-distortion under different distortion metrics is more likely to perform well in most downstream applications.

Second, we asses the effect that the proposed lossy compressor has in SNP calling, as it is one of the most used downstream applications. We use several SNP calling pipelines in our analysis. Furthermore, for the selected individual a consensus set of SNPs exists, which allows us to analyze how accurate the output of the different SNP callers is when the quality scores are replaced by the reconstructed ones.

For the simulations, we set the number of clusters to 3 and 10, and the maximum number of iterations to 50.

4.1 Rate-Distortion results

We analyze the rate-distortion performance of the proposed algorighm and compare it to the following state-of-the-art lossy compressors: PBlock [9], RBlock [9], QVZ [3] and the Illumina binning as performed by DSRC2 [10]. These we chosen based on the results reported in [3]. Moreover, we choose to analyze the performance in term of the six distortion metrics shown in Table 1, as suggested in [3, 9]. Note that while the metrics *MSE*, *L1*, *Lorentzian* are computed on average among all quality scores, the *Chebyshev* and *Max-Min* are metrics that analyze the behavior of the maximum and minimum distortions within a read. The last one, the *Soergel* distortion, is a mixture of the previous two.

Fig. 2 shows the rate-distortion performance with the 30× coverage dataset, for all considered metrics. As can be observed, QVZ is clearly outperformed by RBlock and

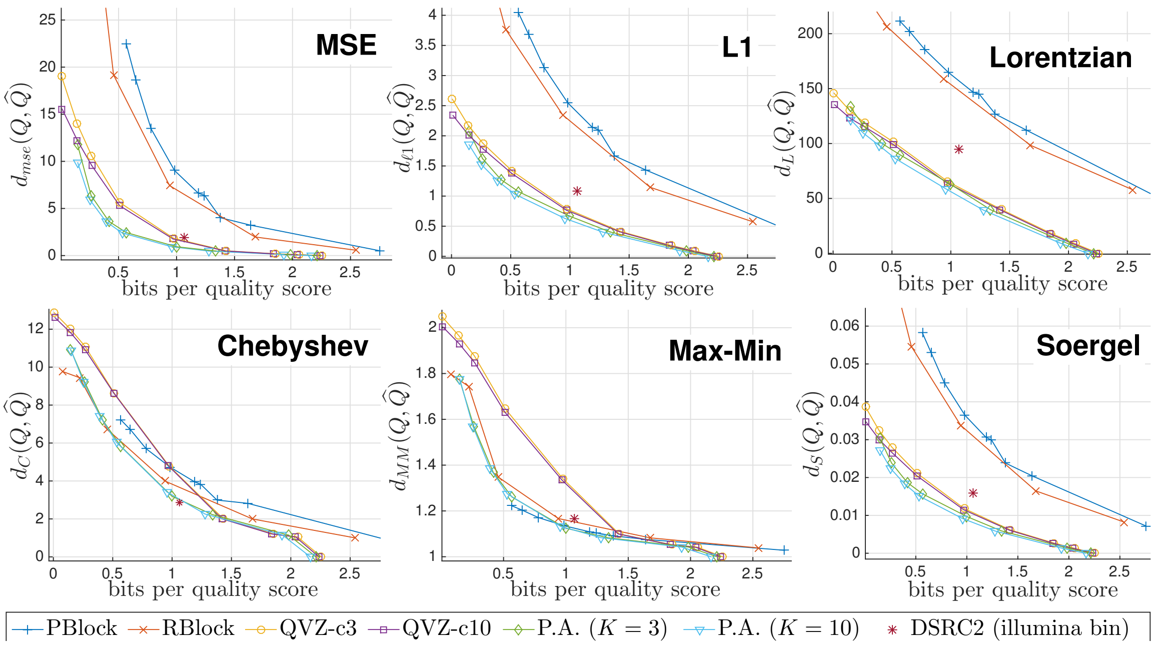

Figure 2: Rate-Distortion results for the 30× coverage dataset. P.A. stands for Proposed Algorithm.

PBlock under all metrics that analyze the maximum distortion within a read (that is, *Chebysev* and *Max-Min*). However, RBlock and PBlock perform poorly unde the remaining metrics. This can be explained by the way in which each of the algorithms operate. Illumina binning generally achieves an intermediate point between them. Thus none of the state-of-the-art algorithms outperforms the rest under all metrics. On the other hand, the proposed algorithm exhibits superior performance under all considered metrics, that is, both those that compute the average distortion across all reads, and those that compute it within a read (with the exception of the *Max-Min* distortion at low rates, where PBlock slightly outperforms the proposed method). For example, for a rate of 1 bit per quality score, the proposed algorithm achieves half the *MSE* distortion incurred by QVZ. And while QVZ is clearly outperformed under the *Max-Min* metric, the proposed method performs better than RBlock and similar to PBlock. This is achieved by an effective clustering step and a careful selection of the distortions levels targeted at each quantizer. Similar results have been observed for the 15× dataset. We have omitted them due to the space constraint.

4.2 SNP calling resuts

Next we analyze how the proposed lossy compressor affects SNP calling. For the analysis, we followed the methodology proposed in [2], which includes the use of the best practices pipelines GATK, Samtools and Platypus (see [2] for details). In brief, we run the aforementioned pipelines with the original data and with that where the quality scores are swapped with the reconstructed ones after lossy compression. Then, the obtained set of SNPs for each case is compared against existing golden standards

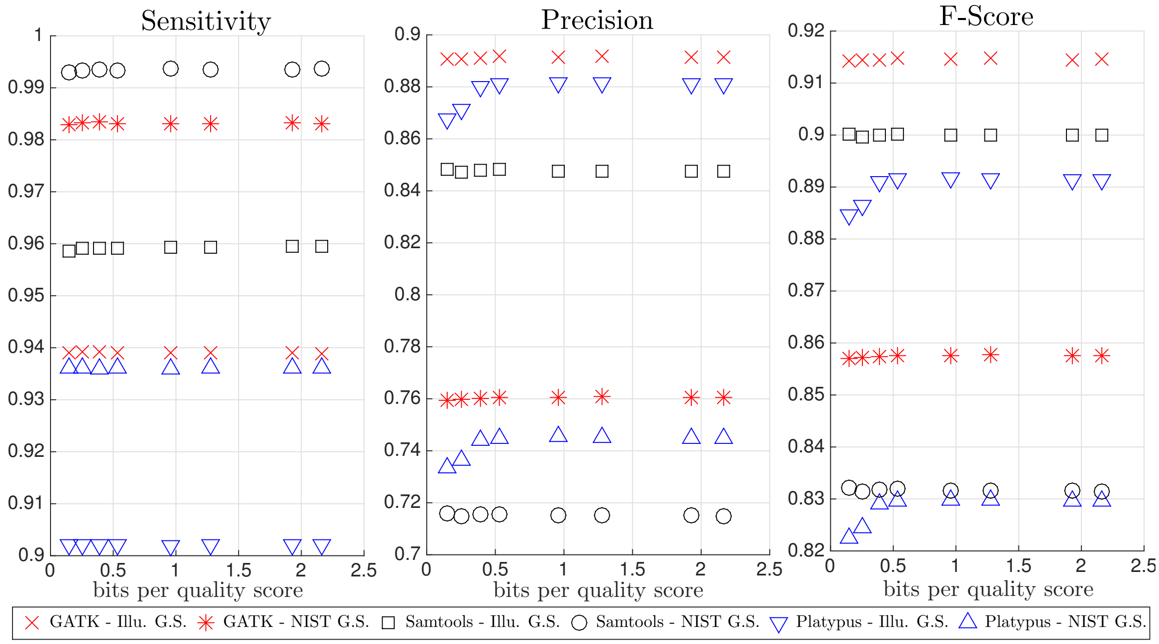

Figure 3: The effect of the proposed lossy compression on Sensitivity, Precision and F-score when performing SNP calling. G.S. stands for Golden Standard. The points with higher bits/quality score correspond to the original data.

(specifically, the NIST standard and the one proposed by Illumina [2]) in order to compute the true positives, false positives and false negatives. Finally, the *specificity, precision* and *f-score* are used for the evaluation criteria.

We emphasize that we are not interested in analyzing how well the different pipelines perform, but in how the proposed lossy compression affects SNP calling. To that end, we define the variability in the output of different SNP calling pipelines as *methodological bias*, and the variability introduced by the lossy compressor within a pipeline as *lossy bias*. Ideally, we would like to show that the lossy bias is orders of magnitude smaller than the methodological bias, as that would indicate that the changes in calling accuracy introduced by the lossy compressor are negligible.

Fig. 3 shows the sensitivity, precision and f-score for the tree pipelines when the golden standard is that of Illumina and NIST. As can be observed, the performance of the different pipelines differs significantly (large methodological bias). Moreover, the performance gets highly affected by the selection of the golden standard. This agrees with the common knowledge that we are far from perfectly calling variants [11]. Most importantly, we see that the variability introduced by the lossy compressor (lossy bias), particularly with GATK and Samtools, is negligible when compared to the methodological bias. For example, the maximum variance within GATK is three orders of magnitude smaller than the difference between the performance of GATK and the other pipelines. Finally, note that even with a very small number of bits/quality score, the performance is in general very similar to that obtained with the original data. These findings suggests that the lossy bias is essentially non-existent relative to the methodological bias, and thus that lossy compression could be used without harming the SNP calling performance.

5 Conclusion

We have proposed a new lossy compressor for quality scores that assumes the sequences are generated by a mixture of time-varying Markov models. Based on this assumption, the algorithm first performs a clustering step, followed by a quantization of the quality scores. The distortion level chosen at each quantizer is chosen to improve the overall rate-distortion performance. Finally, an entropy encoder is applied to the quantized values.

To our knowledge, the proposed lossy compressor is the first to outperform the previously proposed lossy compressors in rate-distortion performance for all considered distortions. Moreover, we analyze how the proposed algorithm affects SNP calling, one of the most used downstream applications for biological inference. We find that the variability introduced by the proposed method is orders of magnitude smaller than the inherent variability that exists across SNP callers. Moreover, for small distortions, the proposed lossy compressor produces more accurate SNP calls than the original dataset. These results are consistent with others in the recent literature suggesting that lossy compression, done judiciously, can be used to boost compression performance without harming and sometimes even boosting the accuracy of the calls. However, we acknowledge the need for performing more extensive studies for substantiating the universality of this phenomenon and consolidating our understanding of it.

References

[1] J. K. Bonfield and M. V. Mahoney, "Compression of fastq and sam format sequencing data," *PloS one*, vol. 8, no. 3, 2013.

[2] I. Ochoa, M. Hernaez, R. Goldfeder, T. Weissman, and E. Ashley, "Effect of lossy compression of quality scores on variant calling," *bioRxiv, http://dx.doi.org/10.1101/029843*, 2015.

[3] G. Malysa, M. Hernaez, I. Ochoa, M. Rao, K. Ganesan, and T. Weissman, "Qvz: lossy compression of quality values," *Bioinformatics*, p. btv330, 2015.

[4] I. Ochoa and *et al.*, "Qualcomp: a new lossy compressor for quality scores based on rate distortion theory," *BMC bioinformatics*, vol. 14, no. 1, 2013.

[5] Y. W. Yu, D. Yorukoglu, and B. Berger, "Traversing the k-mer landscape of ngs read datasets for quality score sparsification," in *Research in Comp. Molecular Bio.*, 2014.

[6] K. P. Murphy, "Machine learning: A probabilistic perspective," *The MIT Press*, 2012.

[7] S. Lloyd, "Least squares quantization in pcm," *IEEE T. on Information Theory*, 1982.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* Wiley, 2006.

[9] R. Cánovas, A. Moffat, and A. Turpin, "Lossy compression of quality scores in genomic data," *Bioinformatics*, 2014.

[10] Ł. Roguski and S. Deorowicz, "Dsrc 2—industry-oriented compression of fastq files," *Bioinformatics*, vol. 30, no. 15, pp. 2213–2215, 2014.

[11] H. Li, "Toward better understanding of artifacts in variant calling from high-coverage samples," *Bioinformatics*, vol. 30, no. 20, pp. 2843–2851, 2014.