On the relevance of quality score metadata in genomic sequence data for omics applications

Thèse N° 9812

Présentée le 12 décembre 2019

à la Faculté des sciences et techniques de l'ingénieur
Groupe SCI STI MM
Programme doctoral en informatique et communications

pour l'obtention du grade de Docteur ès Sciences

par

Ana Angelica HERNANDEZ LOPEZ

Acceptée sur proposition du jury

Prof. A. P. Burg, président du jury
Dr. M. Mattavelli, directeur de thèse
Dr P. Ribeca, rapporteur
Prof. F. Prêteux, rapporteuse
Dr J.-M. Vesin, rapporteur

2019

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Abstract

High-throughput sequencing of DNA molecules has revolutionized biomedical research by enabling the quantitative analysis of the genome to study its function, structure and dynamics. It is driving sequencing-based experiments in life sciences as evidenced by the plethora of emergent omics applications powered by sequence data. However, the capacity to generate massive datasets of sequence data greatly outpaces our ability to analyze them, the notorious bottleneck in omics analyses. With the democratization of computational analyses, practical solutions to the storage, distribution and processing of sequence data will become a necessity for the progress of life science research.

The intrinsic high entropy metadata, known as quality scores, is largely the cause of the substantial size of sequence data files. Despite several efforts to evidence marginal impact on downstream analyses following their lossy representation, no consensus on the limits of "safe" representation with losses exists.

In this research work, we study the effect of lossy quality score representation on three applications: variant calling, gene expression and sequence alignment, to assess the relevance of this metadata for omics analyses. We confirmed negligible impact and discovered that it is possible to compute a threshold value for transparent quality score distortion in sequence alignment, allowing the identification of a "safe" representation for the quality score scale. These results align with current trends in sequencing platforms pushing for coarser resolutions to reduce the storage footprint of sequence data.

Keywords: High-throughput sequencing; genomic sequence metadata; quality scores; variant calling; gene expression; sequence alignment; lossy compression of quality scores; omics.

Résumé

Le séquençage à haut débit de l'ADN a révolutionné la recherche biomédicale en permettant l'analyse quantitative du génome pour étudier sa fonction, sa structure et sa dynamique. Il est la force motrice derrière la pléthore d'expériences omiques émergentes dans la recherche en sciences de la vie. Cependant, la capacité de générer des quantités massives de données de séquences dépasse largement notre capacité à les analyser : c'est le fameux goulot d'étranglement dans les analyses omiques. Les progrès futurs de la recherche en sciences de la vie dépendront de la démocratisation des analyses computationnelles pour le stockage, la distribution et le traitement pratique des données de séquences génomiques.

Les métadonnées intrinsèques à haute entropie, appelées les scores de qualité, sont ce qui provoque la taille importante des fichiers de données de séquences. Malgré plusieurs efforts pour mettre en évidence un impact marginal sur les applications génomiques quand les scores de qualité sont représentées avec perte, il n'existe pas de consensus sur les limites d'une représentation avec perte qui soit "sûre."

Dans ce travail de recherche, nous étudions l'effet de la représentation des scores de qualité avec perte sur trois applications : la détections de variants, l'expression de gènes et l'alignement de séquences, pour évaluer la pertinence de ces métadonnées dans les analyses omiques. Nous confirmons un impact négligeable et avons découvert qu'il est possible de calculer une valeur seuil de la distorsion transparente des scores de qualité pour l'alignement des séquences, permettant l'identification d'une représentation "sûre" pour l'échelle de scores de qualité. Ces résultats s'alignent avec les tendances actuelles des plateformes de séquençage qui préconisent des résolutions plus grossières pour réduire l'empreinte du stockage des données de séquences.

Mots-clés : Séquençage à haut débit ; métadonnées de séquence génomique ; scores de qualité ; détections de variants ; expression de gènes ; alignement de séquences ; compression avec perte des scores de qualité ; omiques.

Contents

List of Figures

List of Tables

1 Introduction

An unprecedented amount of data is being generated at an extraordinary pace. In 2012, it was estimated that 90% of all the data that existed in our entire history had been created in the previous 2 years [1, 2]. Amounting to a total of 2.7 zettabytes (ZB) of digital information, in the same year it was forecasted the generation of five exabytes (EB) of data every two days [3]. By 2013, the digital universe reached 4.4 ZB and its exponential growth rate was observed to account for a doubling in size every two years, projecting the figure for digital data to 44 ZB in 2020 [4].

This data deluge consist of complex data sets that are difficult to process and carry along its volume unparalleled challenges. Big Data alludes to such deluge of data, whether structured, semistructured or unstructured [5], it is produced massively and continuously, and it is fine-grained in scope [6]. The application of traditional processing methods cannot be applied anymore, switching from model-driven to data-driven analysis to investigate these noisy, heterogeneous and voluminous datasets [7].

The Information and Communication Technology industry (ICT) has seen the rapid rise of Big Data, particularly in the domain of the Internet of Things (IoT). According to the International Data Corporation (IDC), the IoT is outpacing the growth of traditional ICT and will soon subsume it. And with mobility and internet connection increasingly featuring prominently in

IoT devices [8], mobile "connected things" will become a key driver of digital data. Considering the ubiquity of devices connected to the internet, and our interest, increasing demand and necessity to use them, it is rather straightforward to imagine fast and intensive production of digital data coming from these sources. According to a recent update to the IDC worldwide semiannual IoT spending guide, the industries that will see the fastest growth through the forecast period 2017-2022 are insurance, government and healthcare [9]. The IDC also estimates that by 2020, close to one third of all Big Data will be generated by the IoT sector [4], whose output is often in the form of text, audio, images or video, a type of data referred as unstructured information [10].

1.1 Big data in genomics

To this day, the type of data generated more rapidly is unstructured, with nearly 95% of existing data being unstructured [5]. This data type is characterized by "human information" [11]: text, photos, movies, internet data (from email, social networks, etc.), scientific simulations, seismic data, genomic datasets, etc., and is coming from three sources, according to the United Nations Economic Commission for Europe's (UNECE) classification for Big Data [12].

However, regardless of the industry sector or source of origin, several domains are leading the production of Big Data. Based on the investigation of the components of the "life cycle" of a dataset (acquisition, storage, distribution and analysis), and as per projections to the year 2025, the domains identified as major generators of Big Data are four: Astronomy, Twitter, YouTube and Genomics. The estimation is that Genomics is either on par with or the most demanding of the four domains [13].

1.1.1 Genomic sequence data

The combination of fast-paced technology, along with highly distributed modes of data acquisition, advances in molecular biology and the coming into prominence of computational biology, could help elucidate the forces behind the domain of Genomics. However, it was

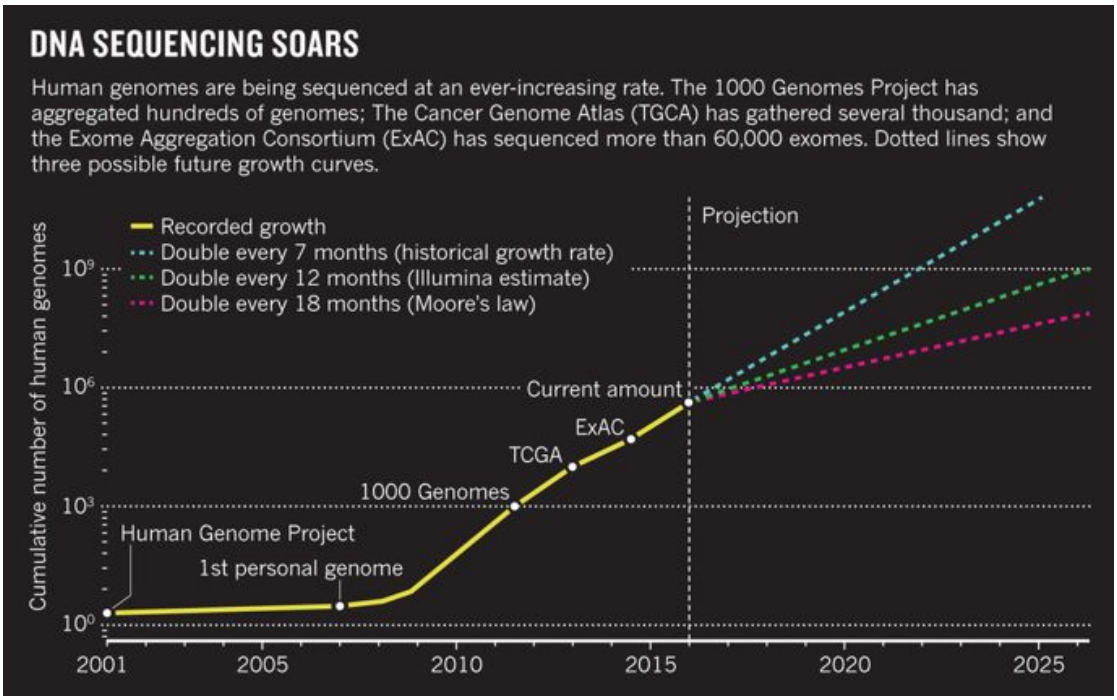


Figure 1.1 – Genomic big data [14].

the advent of genome sequencing, but most notably the introduction of massively parallel sequencing platforms starting from 2004, the so-called Next Generation Sequencing-era [15], that is driving the data flood in Genomics. This data is commonly referred to as genomic sequence data or simply sequencing/sequence data.

Healthcare big data comprises structured data (electronic healthcare records), semistructured data (clinical or administrative messages under global health data standards), unstructured data (clinical notes, medical images, genomic sequence data, etc), and other types of data [16]. In 2011, the health care data alone in the United States was reported to be in the order of 150 EB [17, 18]. It is unclear however the proportion of which corresponds to sequence data. In contrast, it has been reverberated in the scientific literature an estimate figure of 25 EB of worldwide digital healthcare big data by the year 2020 [19, 20], and similarly, without clear distinctions on the figure representing that of sequence data.

Nevertheless, and regardless of precise numbers, the speed at which sequence data has been generated is unquestionably unprecedented (Figure 1.1). The drop of sequencing costs [21]

has by itself been an important driver of genomic sequence data production, along with the arrival of Next Generation Sequencing technologies. See Figure 1.2.

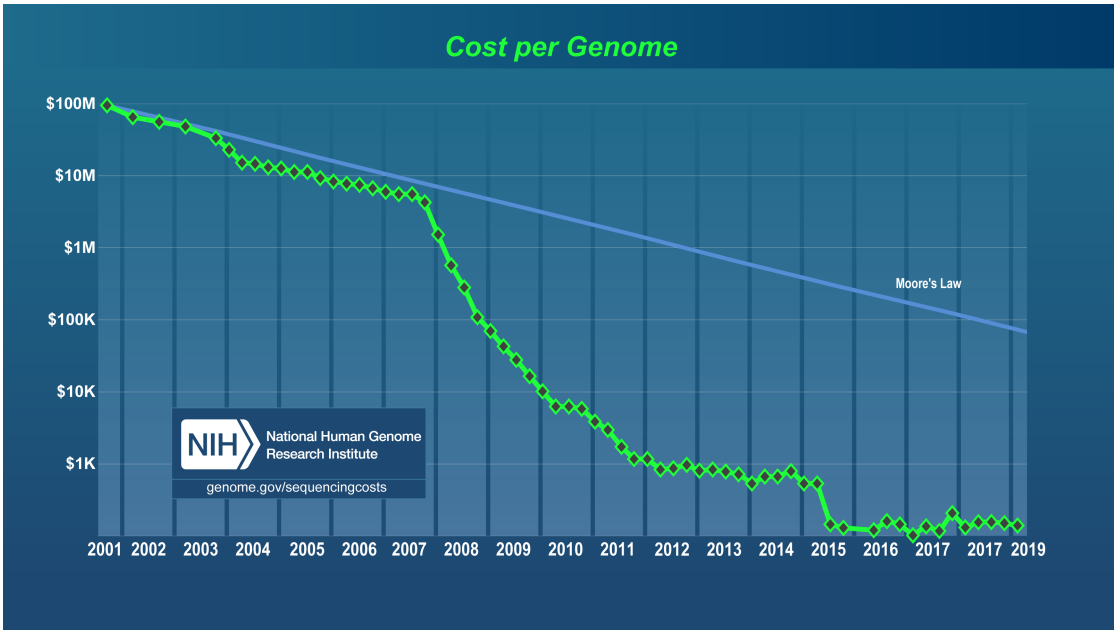


Figure 1.2 – DNA sequencing costs [21].

The historical growth of sequence and resequence data produced worldwide has been approximately doubling every seven months. For sequence data it has been projected an annual storage need between 2-40 EB per year, which conservatively speaking, is on par with the storage estimates for any of the other major Big Data producers, as per [13]: Astronomy (1 EB/year), Twitter (0.001-0.017 EB/year) and YouTube (1-2 EB/year). And with the promise of precision medicine to revolutionize the diagnosis and treatment of diseases, it is reasonable to think of the possibility to sequence an important proportion of the human population in the near future, as per estimates of The Global Alliance for Genomics and Health, more than 60 million patients will have their genome sequenced by 2025 [22]. Consequently exceeding by large the storage growth for the three other Big Data domains.

An example is the 100 000 Genomes Project, a study launched in 2012 in the UK that sequenced one hundred thousand human genomes from patients with rare diseases and their families and patients with cancer [23]. Moreover, it is important to note the competition of private companies to offer genome sequencing services at a population scale with milestones to

reduce sequencing costs, currently pushing to reach the cost of 400 dollars per re-sequenced genome. The era of ubiquitous integration of personal genomic information into aspects of everyday life, or the era of the "social genome", is around the corner [24]. See Figure 1.3.

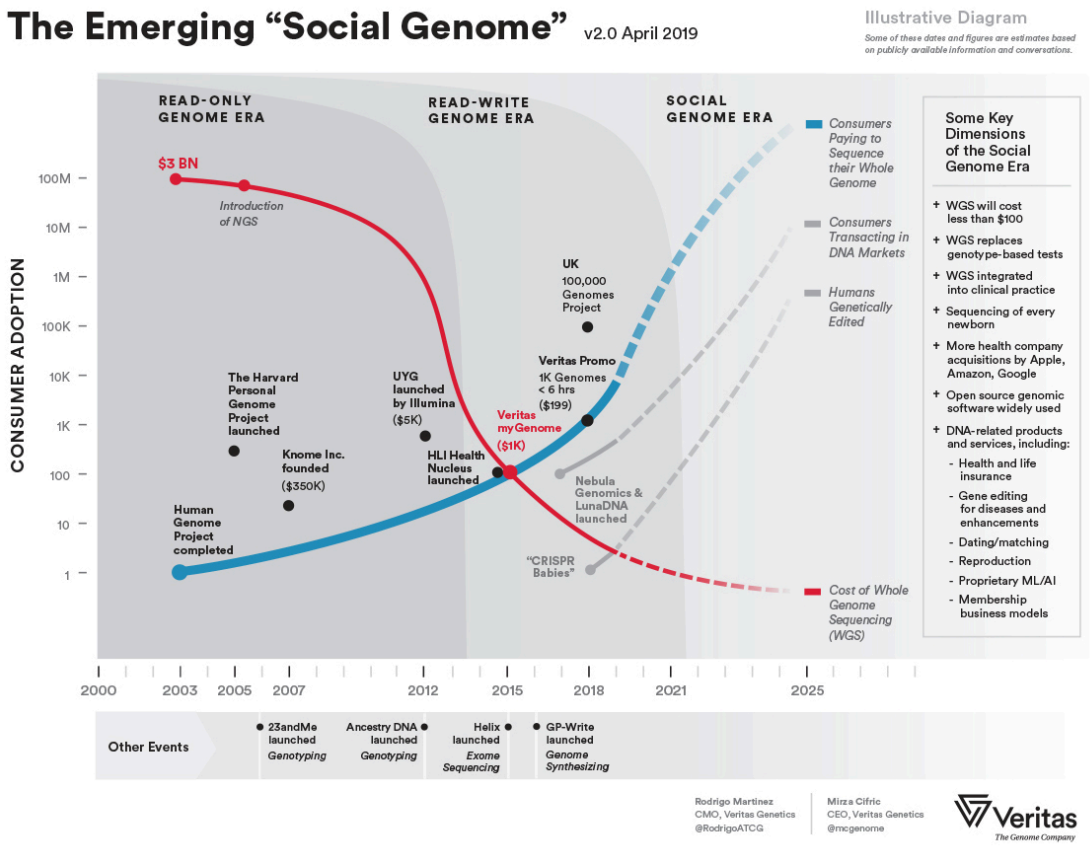


Figure 1.3 – Social genome era (taken from https://www.veritasgenetics.com).

Sequence data storage requirements are tightly dependent on the target application (for example, detection of mutations in a gene) and sequencing assay or preparation protocol (for example, whole genome sequencing). An increasinng number of applications and sequencing assays are discovered every day [25, 26, 27], and the order of magnitude to store only a single sequenced genome can easily range from giga to terabytes [7, 13, 27], pushing with ease several terabytes as data processing and analysis begin [7, 28].

1.1.2 Genomical challenges of sequence data

Researchers are already facing substantial problems to store, manipulate, analyze and generally manage genomic sequence data. There is an ongoing discussion in the scientific literature on how to approach the difficult challenges posed by it, without shortage of recommendations. From the conception of powerful, more mature, and scalable algorithms [29], tools and infrastructures [27] to the reduction or straight out elimination of sequence data, as sequencing accuracy improves [13], thus limiting both the data space needed for the information that we already have and the new information we get by throwing away unnecessary information [30]. And while wiping out sequence data is not expected to be an actionable solution to the storage challenges for genome sequencing, not in the immediate future at least, the field of Genomics will benefit from the lessons learned in particle physics, where raw data is rapidly discarded after acquisition, favoring compressed data summaries [13].

It is conceivably the most pressing challenge to reduce the size of sequence data for storage, as this data is accumulating very rapidly, and the obvious starting point was to explore compression of genomic sequence data [31]. It has been estimated the variation between two human genomes to be in the order of 0.5% when comparing corresponding nucleotide bases (i.e., each element in the string of symbols that makes up the genome) [32]. And while individual genomes are not very compressible, exploiting this kind of intrinsic redundancy allows for groups of related genomes to become highly compressible [29].

Generally speaking, computational analyses over sequence data require the data to be decompressed, a recognized overhead that we are currently willing to pay. Compressive storage in genomics however will become more sophisticated in the years to come, to allow for more efficient computational techniques without decompressing data first [33].

In the meantime, it is of paramount importance to compress sequence data without losing information that is needed [30]. Currently, the community standards for genomic data storage are human-readable text files of raw -unprocessed- sequence data (FASTQ files), and aligned -processed- raw sequence data (SAM files). Many ad-hoc compression methods have been

proposed to reduce the size of both file formats for storage and transmission [34], along with important efforts toward the improvement of such formats (CRAM files [35]) and recent projects to standardize the representation of genomic data for efficient storage, processing and transmission (MPEG-G standard [36]).

1.1.3 Compression of sequence data

Compression of genomic sequence data, whether from FASTQ or SAM files, pertains to changing the representation of genomic information with the primary goal to reduce storage footprint. In this context, genomic information refers to both raw data (sequence data) and metadata (sequence metadata) obtained from sequencing machines during the acquisition process. The loose term of genomic data compression is commonly used to address compression of sequence data and/or sequence metadata.

Much work has been devoted to the exploration of methods to compress sequence data and metadata without loss of information. Benchmarks evaluating the performance of lossless compressors have shown that there is no one-size-fits-all method, concluding that the approach to compression should be paired with the type of sequence data and the target genomic application [34].

The high entropy content of sequence metadata, also known as quality scores, became the bottleneck for compression. The important observation of substantial storage size devoted to quality scores in lossless compressed files, compared to the storage devoted to sequence data, led to the seminal paper that spearheaded the field of lossy compression for sequence metadata [37].

1.2 Lossy sequence metadata

Illumina, a leading developer and manufacturer of sequencing technologies, whose platforms remain the most widely used sequencing instruments [38], followed up with an assessment for the resolution of sequence metadata [39].

The study of lossy sequence metadata, or lossy quality scores, had been put forward. It was effectively initiated with the investigation of techniques for lossy compression, and the quantification of the effect of lossy representation on a downstream genomic application [35, 40, 41]. The realization that considerably smaller file footprints could be achieved by lossily compressing the portion of FASTQ files that pertains to quality scores, clued in to plausible and promising approaches to substantially reduce the size of genomic files for storage.

For all intents and purposes lossy compression involves loss of information, and a dedicated analysis is required to measure and evaluate the impact caused by its usage. It is in this manner that research on the impact of lossy compression of quality scores originated, presently with attempts to try and systematize its evaluation [42].

1.2.1 Impact of lossy quality score representation

While much effort has been dedicated to explore methods to represent quality scores with losses, a larger problem has arisen as a consequence: trying to understand the effect of what was lost. For almost a decade, research has focused on techniques to compress quality scores, validating performance on a single, particularly complex, application that looks for minuscule variations, currently as low as 0.1% [43], in the genome; it is called variant calling. This is an established application that uses tools that rely on quality scores.

The validation comes from quantifying the effect of lossy quality scores in the identification of these tiny variations. But to what point the validation metrics serve well as a proxy for biological significance? For example, what is the implication of reducing performance precision from 99.9% to 99.8% ? Further, is this drop in precision value acceptable? It is then reasonable to ask if the criteria to asses the impact measure in fact the observed effect, so that we can then try and explain the actual effect produced by using lossy quality scores.

1.2.2 Challenges in impact analysis for lossy quality scores

It is well known that artifacts from sequencing, sample preparation and other sources of error, confound the interpretation of results [43] and encumber processing in genomic pipelines. In studying the effect of lossy compression it is sensible to question why research on the subject commonly attributes its application as the main, if not the only, source of impact to output results. For all intents and purposes, lossy quality scores add noise to the above inherent sources of noise, effectively becoming indistinguishable from them. Moreover, the intricate chain of bioinformatic tools that build genomic pipelines [28, 44] only add to their complexity, clouding any intuition that could be derived to explain their collective operation.

Furthermore, it has been been well studied and shown at length, that any lossy approach provide significant storage saving with negligible impact on variant calling [42, 45, 46]. And the conclusion remains fairly the same for simpler techniques for lossy compression. It begs the question, consequently, where to draw the line between good compression, simplicity of the approach, and the effect on variant calling. This tradeoff remains unclear but we strongly suspect the decision is likely to depend on the specific needs for the study at hand. In practice, recent initiatives to standardize pipelines are convening research in several fields to agree on settings to utilize applications, like variant calling, consistently [47]. In the same line, systematic benchmarking strategies are being put forward [48] as modern biology research is increasingly depending on computational omics tools. Their steep development calls for principled assessments of the methods implemented by such tools for more reproducible research and transparency of results.

The pursuit to conduct research on the effect of lossy quality scores in variant calling, albeit the complex pipeline and the need for a high-confidence ground truth, has been justified by the claim to be the most used application for clinical decision making [46]. In addition, the methods and corresponding tools for calling variants depend on quality score values.

In the last years an abundance of protocols and tools have been developed for analysis of RNA sequencing (RNA-seq), by far the most cited sequencing method [26, 49], and as evident

in today's most exhaustive metadatabase for omics tools [50]. Yet, applications for RNA-seq remain to be explored.

In practice, general purpose off-the-shelf lossless compressor like gzip or bzip2 are the de facto standard for raw sequence data. Despite evidence of the application of lossy quality scores to alleviate storage footprints, it is the case that lossy approaches have largely been overlooked and their adoption has yet to take place [51]. Anecdotally however, our observation is that many tools that process sequence data are moving away from using quality scores, leveraging instead information from other sources.

It is possible to hypothesize that the prototypic ad-hoc nature of lossy compressors evoke reticence in the community to their use, in the light of adding overhead and irreversibly loosing information. However, a more convincing argument is the evidence of the emergence of new standards aimed to allow different research groups to produce functionally equivalent results for variant calling [47]. These standards are based on extensive prior work in several domains, including sequence compression. Interestingly, these standards propose the adoption of simple schemes for the quality scores and reduce their representation from 40+ to only 4 levels [47].

In addition, recent advances in sequencing technology are allowing the production of longer genomic sequences with better accuracy and drastically reduced resolution for the quality scores [52]. With such accelerated technological progress, research in lossy quality score compression is rapidly being outpaced by innovation.

1.2.3 Revisiting central questions

It is useful to revisit the questions initiated in 2011 that spurred the investigation of lossy quality scores in genomic sequence data compression [37] to see how they have stood the test of time:

1. Can the quality scores be discarded? The answer back then was "no", with predictions

on improvements in sequencing that would make quality scores largely irrelevant. Today, we attest to the speed at which sequencing evolves with manufacturers, most notably Illumina, pushing a policy to expedite the reduction of quality score resolution by coarse quantization [39, 52]. So we can say that currently quality scores are partially discarded. And while there is not a definite consensus with regard to the irrelevance of quality scores per se, as sequencing technology keeps pushing the envelope, we foresee their irrelevance to come in time, at least to some extent, sooner than later.

2. Is the downstream application robust to small changes in quality scores? At the time it was noted that many applications considered their use critical for inference, such that they would not take sequence data without quality scores. Today, in a way, we have experienced the opposite, finding tools that are conceived to optionally use quality score data or to straight out disregard it. As for robustness of the application to changes in quality scores, currently only documented for variant calling, even large changes seem to produce little impact on the result.

1.2.4 Open opportunities for impact analysis of lossy quality scores

We have identified in the literature the following shortcomings:

- The holistic approach to quantifying impact lossy quality scores in the light of evidence of confounding measurements

  Measuring only the cumulative effect of lossy quality scores in pipelines for variant calling obscures understanding of their precise effect. In the literature, the identification and analysis of processing steps in the pipeline chain that are susceptible to lossy quality scores, have been consistently overlooked. Individual contribution of these steps is a source of variation whose effect is passed along the pipeline, and is ultimately reflected cumulatively in the result. This is specially important considering variant calling is ultimately looking for variations in data with small frequencies [53]. Sources of variation from computational tools are present in fundamental steps and also unanticipated

variations have been detected throughout the pipeline [54, 55, 56].

- Absence of detailed pipeline systematization in analyzing impact of lossy quality scores

  Variant calling is an intricate downstream application that requires a meticulous succession of tools carefully configured for processing. The choice of tools and their setup impact the computation of the expected result, as it has been reported lately [56]. To reduce variability in core pipeline components and to harmonize upstream steps prior to the core variant calling step, data processing standards were proposed last year [47]. The goal is to have the capacity to run two pipelines independently on the same data to produce two output files that, upon analysis by the same variant caller, produce the same result. The advice given in laying out a testbed for impact analysis for lossy quality scores is to follow recommended best practices for variant calling. These rules however miss out on values to configure tools, selection of reference files, and choice of tools, which is not always precise. As a result, there is a lack of uniformity in setting variant calling pipelines as testbeds for impact analysis, as we discovered in going through the supplementary information of several scientific papers, for example [42, 57].

- Narrow scope of impact evaluation to a single downstream application

  Much work has been devoted to the exploration of variant calling as a testbed to evaluate the effect of lossy quality scores. A look into today's available sequencing methods suitable for an increasing number of downstream applications [49, 58], and accompanied omics tools, calls to broaden this exploration. Moreover, with the prevalence of multiomics experiments leveraging genomic information from multiple datasets, and the associated challenge in storing and managing these sequence data, the pertinence of quality scores is at stake.

As it was prudently noticed early on [40], it is easy to reduce the size of quality scores by any lossy method but it is very hard to determine the effect such transformations will have on downstream analyses.

Reducing the size of sequence metadata is the obvious course of action to immediately aid

to alleviate sequence data storage. To this purpose, several lossy quality score compressors have been devised without getting much traction to their utilization. Perhaps their versatility in taking in any type of sequence data is overshadowed by their general applicability in downstream analyses. Given the delicate measures that are desired to infer in the complex ad-hoc omics pipelines, one could posit such generality does not serve them well. And since there is no particular intuition from conception on how the lossy representation would impact subsequent analyses, there are lack of guarantees, which contributes to limiting their adoption.

Thus the need now is not to develop more lossy compressors for quality scores. As we see it instead, the need is toward evaluating the pertinence of lossy quality score representation in the context of downstream applications. Ultimately shedding light on the effect of their usage to quantify their relevance.

1.3 Purpose statement and thesis contributions

In pursuit of addressing the shortcomings stated in section 1.2.4, the purpose of this study is to investigate the relevance of quality scores, as per the collateral effect they pose when representing them lossy for storage savings, on selected omics applications.

When we started our investigation we discovered the impracticality of inspecting susceptible processing steps to lossy quality scores in omics applications. We started by focusing on variant calling, noticing there was no consensus on a methodology for the evaluation of impact analysis. In addition, we were confronted by the paradox of choice after looking at the sheer number of computational methods available to build a pipeline for this application. As for 2017, more than 40 open-source tools were available just for variant calling [53]. Today, there exist 160, as per Omicstools' database [50], an evidence to the speed of omics tools' development.

To organize and facilitate future evaluation for the effect of lossy quality scores, we gathered strategies, tools and pipelines commonly used in the state-of-the-art to analyze the impact on variant calling. Along with this, and toward a systematic assessment of impact analysis, we

put together a benchmark with the intention to be used as future reference to evaluate lossy compression tools for the quality scores in variant calling.

We then branch out to investigate the utilization of quality scores in other omics applications, tools and pipelines. With RNA sequencing becoming and area of much ongoing research and innovation, we decided to explore applications based on this sequencing method. In omics applications complex pipelines are the rule rather than the exception, and their analysis is certainly challenging. Keeping in mind our goal to evaluate the impact of lossy representation of quality scores, we proposed a testbed for their analysis in differential gene expression, what is perhaps after variant calling, the most researched omics application. We devised a pipeline that streamlined the chain of processing steps, while still taking into account the quality scores, and proposed a strategy to quantify the effect of using lossy quality scores in this application.

With the knowledge acquired from the benchmark for variant calling and the streamlined pipeline for gene expression, we identified a candidate element whose role we acknowledged relevant to analyze the effect of lossy quality score representation: sequence alignment. This core element is a fundamental processing step to all pipelines for omics applications, and is itself a much researched application whose role plays out importantly in processing pipelines. Research in the las couple of years on the subject has started to emerge remarking the impact sequence alignment alone has on variant calling [56, 59].

We focused then on sequence alignment and investigated a particular tool, suitable for both variant calling and gene expression, to quantify the effect of lossy quality score representation. We discovered that it is possible to identify a threshold for transparent lossy quality score compression without loss of accuracy. And conveniently, we picked up where the state-of-the-art left off several years ago, as per suggested future work, it "should concentrate on studying lossy quality score compression, strictly guided by minimizing loss of accuracy in alignment, SNP calling and other applications" [40].

In summary, the investigation reported in this thesis contributes to the state-of-the-art with:

- A benchmark to evaluate the impact of lossy compressing quality scores in the omics application variant calling
- A testbed for the evaluation of lossy compression of quality scores in the omics application based on RNA-sequencing, differential gene expression
- A lossy representation of quality scores for transparent compression along with the identification of a transparency threshold, in the omics application sequence alignment

1.4 Thesis statement and thesis organization

Thesis statement:

> Lossy representation of quality scores in omics applications allows for significant reduction of sequence data storage footprint with the caveat of uncertain impact following their usage. It is possible to circumvent this limitation, and to transparently represent lossy quality scores in streamlined omics applications, while giving guarantees of collateral impact following their application.

This thesis is organized in six chapters centered on the contributions stated in section 1.3. Chapter 2 reviews the state-of-the-art and focuses on genome sequencing technologies, quality score representation and lossy compressors for quality scores. Chapter 3 delves into the omics application variant calling, and describes the evaluation benchmark to assess the effect of lossy quality score representation. Chapter 4 examines the omics application differential gene expression, describes the organization of a streamlined pipeline, and presents a method to assess and quantify the effect of lossy quality score compression. In Chapter 5 we explore sequence alignment, devise an approach to organize and quantify alignment results, and show how to leverage an alignment tool for transparent representation of quality scores. We conclude in Chapter 6 with the final remarks.

2 State of the art

The organization of nucleic acids in chains describe the genetic and biochemical information that supports life. The discovery of the tridimensional structure of DNA in 1953 was the cornerstone to the development of a conceptual framework to understand the composition of living matter. The capacity to read out the content of DNA has been made possible by DNA sequencing, which has played a fundamental role in the analysis of genomic sequences of organisms to discover their structure, organization and function.

The field of genomics was born in the late 1970's, and its scope was originally the study of the structure and function of genes, as well as hereditary and evolutionary relationships inter and intra species [60]. The term "genomics" is used somehow loosely now with some implied meanings to what is currently attributed the neologism "omics". The word and suffix omics alludes to the quantification of biological molecules of living organism through their nucleotide sequences, in order to study their function, organization and dynamics. As suffix, it encompasses an ever-growing number of fields such as transcriptomics, proteomics, metabolomics, epigenomics, nutrigenomics, evolomics, systeomics, for example.

2.1 Sequencing technologies

A sequencing machine outputs files with DNA sequences, the genomic data, represented by strings of symbols called nucleotides or bases. They are elements from a four level alphabet that stand for each possible letter in the organization of the DNA, which are: A, C, G and T. An additional symbol to denote ambiguity is also part of the alphabet, thus it is actually a five letter alphabet. The first technology for sequencing was developed in 1977 by Sanger and Maxam [61, 62], and it became the most applied technique for sequencing [63], dominating for over thirty years. The technologies used in this period are referred as the first-generation of sequencing [64]. The development of more efficient and faster technologies followed, motivating the creation of centralized repositories to collect the sequence data like GenBank. From its beginnings in 1982, the sequence repository growth doubles every 18 months[1].

In the year 2000, a new era of sequencing technologies followed with the arrival of machines capable to provide massive parallel throughput at a much lower cost, the so called high-throughput or next-generation sequencing [65]. In this type of sequencing reactions occur in parallel, and are spatially separated on a solid surface [66, 67]. The limitation of these technologies is the production of millions of short sequences of DNA instead of a complete sequence of the full genome.

Following the first generation, next-generation sequencing is further divided in two generations: the second- and third- generation sequencing. According to [60], the second-generation is characterized by the need to amplify libraries, in contrast to the most recent third-generation that needs not. There is considerable discussion about the defining characteristics of each but as it is remarked in [68], there seems to be a consensus in that third-generation technologies are capable of sequencing single molecules without the requirement of DNA amplification, which is shared by the previous technology.

The process of amplification enabled the production of multiple sequences at the cost of introducing base sequence errors, and favoring certain sequences over others, changing their

[1] https://www.ncbi.nlm.nih.gov/genbank/statistics/

relative frequency and abundance [69]. With the third-generation, the sequencing from a single DNA molecule without the need for amplification was possible, as it was also the fast production of longer reads, albeit with very high error rates [70], at lower cost. This generation is also called single-molecule sequencing.

The first-generation sequencing approaches lasted for three decades and their limitations in cost and time were largely improved by next-generation sequencing in the following fronts [63]: (i) the parallel generation of millions of short read sequences; (ii) the speed of the sequencing process compared to the previous generation, and (iii) the drop in cost.

In the second-generation, short-read sequencing methods are grouped under two approaches: sequencing by synthesis and by ligation [65]. The main sequencing platforms are Roche/454, Illumina/Solexa, and ABI/SOLiD; Illumina is currently the dominant supplier of sequencing instruments [70]. These platforms can generate raw sequence bases in the order of five hundred million to billions of bases in a single run [69].

The technology for the third-generation provides the following key advantages over the previous one: (i) higher throughput; (ii) faster turnaround time (high coverage sequencing in minutes); (iii) longer read lengths; (iv) higher consensus accuracy for detection of rare variants, and (v) lower cost. The most prevalent sequencing platforms in the third-generation are Pacific Biosciences and Oxford Nanopore [63].

Next-generation sequencing keeps evolving with no signs of plateauing in cost or throughput. and the exploration of new approaches to sequencing, for example using quantum tunneling [71], graphene nanopores [72], or by reading out nucleic acids directly in fixed cells [73], carries on.

2.2 Outlook and challenges

Since the end of the Human Genome Project in 2003, and with the introduction of next-generation sequencing platforms in 2005, the scientific community has launched and navi-

gated ambitious projects, as evident by the landscape of scientific advances this technology has enabled. The vision for genomics is that the most effective way to improve human health is through understanding genome biology as a basis for understanding disease biology, which then becomes the foundation for improving health [74].

The increasing affordability of next-generation sequencing is enabling many applications to study the genome. Notably, de novo assembly to piece together genomic sequences to get a first draft of complete genomes; a lengthly process that combined short read sequences to be assembled to genomes with repetitive structures. However, with the availability of longer reads and longer range contiguity information, it is now within reach genome assemblies of acceptable quality [70]. The scope and range of applications for genome sequencing is shaped by the underlying technology, and both continue to expand. Key areas of applications include whole-genome resequencing (targeted sequencing), RNA sequencing, genomic variation and detection, profiling of epigenetic marks, chromatin structure and personal genomics [69].

The challenges of sequence data storage, analysis and interpretation are now fundamental constraining factors limiting the use of next-generation sequencing. When working with sequence data, four levels of analysis need to be considered [60]:

- The acquisition of sequence reads using the software provided by the manufacturer of the sequencing platform to call nucleotide bases from raw signals in order to produce reads with their associated quality scores

- Assembly or alignment of reads

- Annotation, data integration and visualization

- Combination of data into a processing pipeline

2.3 Storing sequence data

The process of base calling during sequencing produces raw sequence data, short reads with associated per base quality scores. The latter is metadata whose scoring system is platform-specific. Sequence data is commonly stored in the widely adopted FASTQ format [75], although sequencers' native formats are also used.

In the domain of bioinformatics there is a profusion of ad-hoc formats for data manipulation. Those that are successfully adopted become de facto standards, despite being ambiguously defined or burdensome to utilize. The FASTQ format is an example of such standards, it was invented at the Wellcome Trust Sanger Institute, but never formally described. The FASTQ file format was gradually disseminated and evolved by consensus to complement the FASTA format [75]. The extension made to the latter was to incorporate sequences of numeric values, called quality scores, to each read sequence.

2.4 Base quality scores

More than twenty years ago, the software phred was developed to improve the accuracy of base-calling by assigning an error probability to each called base in a sequence read [76]. This introduced the phred quality score of a base call, defined as $Q_{phred} = -10 \times \log_{10}(P_e)$. A high phred quality score implies that a base call is more reliable and less likely to be incorrect. Phred scores are currently a de facto standard for representing the quality of sequence reads.

Phred scores are stored as single characters, and restricted to the ASCII scale of printable characters to originally facilitate their reading and editing [75]. The first FASTQ files (Sanger FASTQ files) encoded phred qualities from 0 to 93, using the full ASCII range (ASCII 33-126). This encoding permitted an ample range of error probability for calling a base: a base being wrong ($P_e = 1$) through it being called very accurately ($P_e = 10^{-9.3}$). The Sanger FASTQ encoding for quality scores is considered as the original or standard phred encoding for the

Phred Quality Score	Probability of Incorrect Base Call	Base Call Accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1,000	99.9%
40	1 in 10,000	99.99%
50	1 in 100,000	99.999%

Figure 2.1 – Quality scores and estimated base calling error.

quality scores in the FASTQ format. The Open Bioinformatics Foundation[2] refers to it as the Sanger standard.

The estimation of errors varies among sequencing systems, who have invented their own versions of incompatible FASTQ formats, contributing to the confusion. However, the Sanger version has, by and large, received the broadest acceptance. In Figure 2.1 the relationship between quality scores and base calling for typical Illumina sequencers is shown [77].

High-throughput sequencing technologies are changing the scenario of genomic information manipulation. The important reduction in sequencing costs in terms of resources and time achieved in the last few years has led to a production of large volumes of genomic data, and the rate at which this information is being generated is rapidly outpacing the physical capacities for storage and transmission. It has thus become increasingly important to look into ways that can enable everyday use of genomic information for large scale applications.

To this day the main efforts in genomic information manipulation are being concentrated in the compression of genomic data. There exists several tools for this purpose and their approaches to compression vary greatly. Their ultimate goal is however to achieve a substantial compression ratio, which is ultimately the most important term of comparison. As we will see in the following chapter, quality scores pose a very real, practical problem for the storage of sequence data.

[2] https://www.open-bio.org/

2.5 Genomic compression

Making the size of genomic data smaller is not only for the purpose of reducing the storage space but it is also to facilitate its distribution. The compression of genomic information is an open challenge that has been approached by different methods that, to this day, achieve modest compression ratios when considering the size of the input data they take in.

The trends in storage, transfer and sequencing call for the exploration of new approaches that take advantage of the deluge of data that has been made available by high throughput sequencing technologies [78]. Sequencing is steeply heading to higher throughputs at reduced costs. This tendency is notoriously steep in comparison to data storage and transfer, a clear indication for the need of efficient compression of genomic information.

Genomic data is represented as a stream of symbols that are read out from a sequencing machine, also known as sequencer. As we have discussed, these readouts, or simply reads, are assigned by the sequencer a confidence level called quality score to quantify the certainty of the read. The reads along with their quality score make up the genomic data. As of today, two prevalent file formats are being used for genomic information. These file formats are FASTQ, for raw data, and SAM/BAM, for aligned data.

In general, genomic compression tools can be classified into two categories, depending on the type of file format they can be fed to. There exists however some tools that support both types of genomic data files. Genomic data for research purposes are published by several organizations. The richest datasets are provided by the 1000 Genome Project[3], the Genome Expression Omnibus repository[4], and the European Nucleotide Archive[5]. These repositories help in the development and test of compression techniques for genomic data.

An initiative to identify a limited set of genomic data publicly available to cover the largest possible extent of sequencing technology and type of experiments was proposed by the MPEG-

[3] http://www.1000genomes.org/
[4] http://www.ncbi.nlm.nih.gov/geo/
[5] https://www.ebi.ac.uk/ena

G group [79]. The initiative, among other things, intends the adoption of the proposed dataset for research purposes.

2.5.1 Compression of sequence reads

Next-generation sequencing has paved the way for the exploration of data modeling in genome compression. This is relevant given the tremendous amount of information contained in genomic data, placing the analysis of their statistics as the next natural approach to achieve high rates of lossless compression. One of the first lossles compressors to capitalize on the particularities of genomic data was CBC [80]. The algorithm uses aligned data, and its approach is based on the construction of specific probability models relevant to the input data, which is described by sequentially processing the symbols in every read to build up a context. This can be understood as a training aimed at determining the distribution of the reads based on previously seen symbols.

Many lossless compressors for sequence data have been developed in the las two decades [34], and they perform rather well in common assessment metrics such as compression ratio, memory usage, and compression and decompression time. The trend now in lossless compression is the combination of techniques to improve ratios [81], and the incorporation of capabilities to add versatility [51].

The realization that the compression ratios achieved with previous approaches are not sufficient when compared to the size of the input data files has steered the research to a thorough examination of the genomic data.

2.5.2 Lossy compression of quality scores

When lossless compression is adopted to limit the storage requirements of sequence data, quality scores account for the largest part of the overall compressed information [40, 46, 81, 82]. While nucleotides strings can be compared to an external known reference genome and differential compression can be applied, with quality scores this is not possible. Moreover, this

metadata has a much wider dynamic range than nucleotide sequences. Research for efficient compression of quality scores has long shifted from the lossless approach, adopted by popular tools such as Samtools [83] and other more optimized implementations [40, 41, 84], to lossy schemes.

It has been noted that the values aimed at measuring the reliability of the sequences readout take up a large chunk of the overall compressed file size, as these values are fine-grained and high in entropy. This important observation has propelled a discussion on whether the quality score values are indeed significant to downstream applications, and whether or not keeping them in their entirety is necessary.

Efforts on quality score compression date back to less than ten years, starting with the seminal paper that proposed the first method to represent them lossily [37]. The tradeoff for losing accuracy of these values is a significant reduction in the size of the compressed file, and the extent to which this loss is permitted should be subject to the application that operates on the genomic data (that is, the reads).

Ideally, a lossy compressor of quality scores should factor in the downstream application but their very specific nature make this task difficult. A workaround to this problem is the use of a flexible metric that allows to measure the distortion of the lossy compression so as to accommodate a desired value of distortion. This is the method followed by the algorithm QVZ [85] whose approach built upon the ideas of QualComp [86], the first compressor to encode lossy quality scores with respect to a flexible distortion metric. QVZ's method is based on the observation that adjacent values of quality scores within a read exhibit strong correlation, a feature that is exploited to compute probabilities of their occurrences, which are used to build a set of quantizers that minimize a given distortion so that every quality score will map to a quantized value.

Drawing from the distinction proposed in [42] on whether lossy compressors use biological information [45, 87] or not, QVZ and P-/R-Block [88] are good representatives of state-of-the-art compressors that do not. While QVZ uses the statistics of the quality scores, P-/R-Block

does not and instead separates quality scores into blocks of variable size, where all quality scores contained in each block comply with a chosen parameter according to some measure criterion. For each block, its length and a representative value are stored.

Quartz [87] leverages on biological information to generate a dictionary of common k-mers for each species. Then, for a given set of sequence reads, the compressor breaks them up into a set of overlapping k-mers. Subsequently, every position in a supporting k-mer different from a dictionary k-mer is annotated as a possible variant. Quartz assumes that divergent bases in supporting k-mers correspond to sequencing errors or single nucleotide polymorphisms (SNPs), and their corresponding quality scores are preserved while the rest are set to a predefined default value.

The current direction in lossy compression, similar to its lossless counterpart, is the support for features to add versatility. In particular, incorporating options for several lossy compression modes [46].

2.6 Standardizing the representation of genomic information

The fast-paced advances in sequencing technologies and widespread reception has led to a flood of massive high-throughput sequence datasets with fundamental operational problems to extract value from them. The increasing computational complexity and costs associated with the storage, transmission, and analysis of sequence data are notoriously becoming the new bottleneck. On top of this, the profusion of ad-hoc data formats and prevalent lack of guidelines and standards in bioinformatic analyses have motivated the emergent open standard for genomic information by the Moving Picture Expert Group (MPEG) and the ISO Technical Committee 276/Working Group 5 [36]. This new standard is already gaining attention in the community [31, 51, 81].

The new open standard, MPEG-G, addresses the limitations of current technologies and sequence data formats for efficient and economical management of genomic information. Capitalizing on MPEG's experience for the creation of lasting standards that enable the in-

teroperability and integration of digital media, MPEG-G has been developed for efficient compression, storage, transmission and processing of sequence data.

The new open standard gives particular importance to the representation of compressed raw and aligned sequence data, and support for storage and transmission through the definition of a transport layer for genomic information. Support for selective access to compressed data, aggregation of studies and incremental update of sequence data, encryption, and other features have also been incorporated.

3 Lossy quality scores and detection of genetic variants

In the last few years over a dozen methods have been proposed to reduce the entropy of quality scores in sequence data. This compact representation comes with the benefit of improved compression and consequent storage saving, but at a cost of introducing distortion. In principle, this new representation for the quality scores is to be looked for to reduce collateral effects that come after, and as a consequence, of its usage. However, the specific and complex nature of omics analyses and the large number thereof, make it rather complicated to foresee from conception the impact a coarse a representation for the quality scores will have over an application. Moreover, the variability of sequence data, along with the intricate succession of processing steps carried out in omics pipelines, add to the difficulty of developing an intuition to leverage on when devising a reduced representation for the quality scores. As a result, the methods implementing these new representations are specific and ad-hoc to raw sequence data. However, their performance is in principle compromised by their generality, in that they are not specific for a particular omics analysis.

In this chapter we motivate the case for lossy quality score representation and present the idea of a testbed to quantify the effect of such representation in omics applications, alongside evidence of the size of quality scores in raw sequence data files. Then, we describe the fundamentals of variant calling, the omics application onto which lossy quality score representation has been evaluated in the state-of-the-art. We follow with a framework that presents datasets,

tools, metrics and a procedure to evaluate the effect of lossy quality score representation over a pipeline for variant calling. We finish with a brief discussion and conclude the chapter.

3.1 High-throughput sequencing and the storage problem

In the last few years the fast-paced advancements in sequencing technology have created new challenges in the domain of genomic information. As an unprecedented amount of data are being made available, the problem is now inclining on storing the data as efficiently as possible. For this purpose, in the last couple of years a good number of new genome compression tools have been developed. High-throughput sequencing technologies are changing the scenario of genomic information manipulation. The important reduction in sequencing costs in terms of resources and time achieved in the last few years has led to a production of large volumes of genomic data, and the rate at which this information is being generated is rapidly outpacing the physical capacities for storage and transmission. It has thus become increasingly important to look into ways that can enable everyday use of genomic information for large scale applications.

In the decade since the completion of the Human Genome Project, genome sequencing technology has undergone advances that have outpaced Moore's Law, and sequencing centers are producing data at an unprecedented rate. The rapid growth of genomic sequencing data has resulted in difficulties in storage and transmission [89].

3.2 Genomic compression to alleviate storage of genomic files

High-throughput genome sequencing machines produce genomic information in the form of strings of nucleotides (bases) and associated metadata. Quality Scores (QS) account for the largest part of the overall compressed information when lossless compression is adopted; one reason for this is their larger alphabet, and greater dynamic range than that of the four nucleotides [82]. A QS is a number output by a sequencing machine signaling the estimating probability that a base is correctly identified by the sequencing process. The use of QS in

downstream analyses is extremely diversified, and their use is dependent on the pipeline, omics tools and application.

The attempt to achieve higher compression rates than those yielded by lossless approaches and other optimized implementations [82, 90] is leading to the study of lossy schemes for QS, as has been reported in the literature [85, 86, 88, 91]. These works have made the observation that, in some cases, lossy representation of QS does not negatively affect the quality of results but seems however to actually improve performance of certain analyses such as variant calling (identification of variants with respect to a reference genome) [87]. On one hand, these conclusions run counter to the conventional knowledge that by simplifying the representation of QS we are discarding information, which would impact the quality of the final result. On the other hand, to alleviate storage and facilitate data manipulation and processing, it seems pertinent to adopt a lossy representation of QS with appropriate constraints so as to minimize the impact on downstream analyses.

In the field of video or audio lossy compression, the solution for the definition of the distortion function, despite several attempts at defining objective distortion metrics, has been to use the perceived visual quality of expert viewers or expert listeners under viewing and listening conditions specified by standard protocols. By these means, coding schemes are compared and ranked at specific bit rates according to the lowest perceived visual or auditory distortion, or for the same perceived distortion, to the lowest bitrate necessary. Although some rate-distortion metrics have been previously proposed [85, 88] for QS, no consensus on appropriate definition of distortion exists in the scientific community. QS metadata are commonly used in variant calling to identify genomic variations, such as single nucleotide polymorphisms (SNPs) and insertions or deletions (INDELs). In other omics applications they are used as an additional source of information to help the mapping of sequences to a reference genome, and to assemble sequences into longer nucleotide strings.

Transforming the original representation of the QS to a coarser granularity reduces their entropy and makes them more compressible. Lossy QS metadata is a new representation for

the QS that results from applying a transformation onto the original QS values (3.1). It follows that we can map this transformation to a measure of "accuracy" of results of the omics analysis when lossy QS are used. In other words, the application of lossy representation of QS in variant calling will induce a result, which includes the effect produced by the lossy representation, and whose overall quantification is represented by the value of the accuracy of the omics analysis.

$$T: QS \rightarrow QS^* \tag{3.1}$$

In the following sections we will define an appropriate methodology for the measure of the "quality" of variant calling analysis results. Such metrics will constitute the base to evaluate the effect of transforming QS metadata and will be used to compare and rank different approaches to lossy compression of genomic sequence metadata.

3.3 Genomic sequence data

Sequence data are produced by high-throughput sequencing machines in the form of strings of symbols, representing nucleotides in molecules of DNA or RNA strands taken from an organism's sample. Streams of nucleotide strings or nucleotide sequences are what we refer to as genomic information, which is the output of sequencing machines or simply sequencers. The process of assigning a given symbol to a position in the sequenced genomic string is called "base calling". Nucleotides are also called bases.

Because nucleotide strings are read out from sequencers they are often called genomic sequences, read sequences, sequence reads or just reads. The bases, along with their confidence value of a correct base call (quality score), are what we refer to as genomic data.

3.4 File formats in sequence data

One of the core issues of Bioinformatics is dealing with a profusion of (often poorly defined or ambiguous) file formats. Some ad-hoc simple human readable formats have over time attained the status of de facto standards [75]. As of today, two formats are used to store both bases and QS. The two file formats are FASTQ [75] and SAM/BAM [90] and the main difference between them lies in a notion of order in which genomic data are stored and described. Under this definition, FASTQ files store raw sequence data coming directly from a sequencing machine, or raw sequence data that has been minimally preprocessed.

Data in FASTQ files are subsequently operated on, analyzed and interpreted to derive meaningful information in line with the purpose of the host omics application. Leading the succession of such processing steps is the alignment of raw sequence reads onto a special sequence known as reference genome, or simply reference sequence, which is a representative nucleic acid sequence of a species. During alignment, sequences are mapped to likely locations from which they originated in the genome. The set of aligned reads describes in detail the content of genomic data that has been sorted with respect to the reference sequence. Aligned reads are stored in SAM/BAM format. See Fig 3.1.

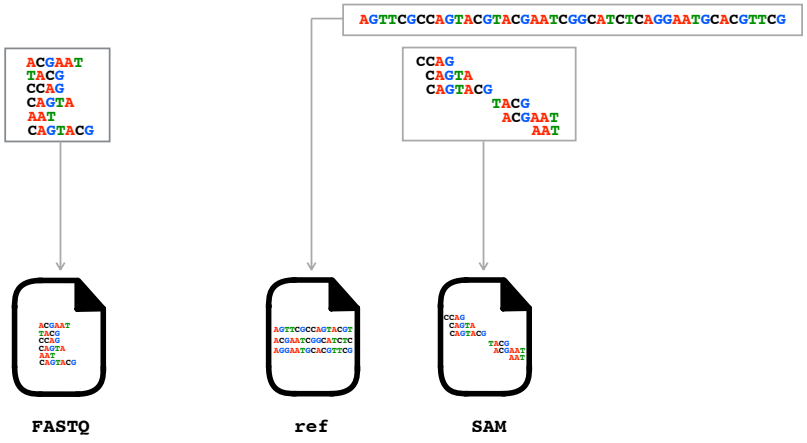


Figure 3.1 – Schematic of the two de facto file formats for genomic data. FASTQ format stores unaligned data. SAM/BAM format stores aligned data to a reference sequence.

3.4.1 FASTQ format

This format extends a simpler one called FASTA by including a numeric value for each base in the sequence read. The FASTA sequence file format originated as an input format for an alignment suite [92] and it is used to store any sort of sequence data that does not require the inclusion of per base quality score.

Customary sequence data stored in FASTA are reference genome files, protein sequences, transcript sequences, and other similar sequences. Each entry in a FASTA file contains two elements: a description header and the nucleotide sequence. Refer to Fig 3.2.

```
>ENSMUSG00000020122|ENSMUST00000138518
CCCTCCTATCATGCTGTCAGTGTATCTCTAAATAGCACTCTCAACCCCCGTGAACTTGGT
TATTAAAAACATGCCCAAAGTCTGGGAGCCAGGGCTGCAGGGAAATACCACAGCCTCAGT
TCATCAAAACAGTTCATTGCCCAAAATGTTCTCAGCTGCAGCTTTCATGAGGTAACTCCA
GGGCCCACCTGTTCTCTGGT
>ENSMUSG00000020122|ENSMUST00000125984
GAGTCAGGTTGAAGCTGCCCTGAACACTACAGAGAAGAGAGGCCTTGGTGTCCTGTTGTC
TCCAGAACCCCAATATGTCTTGTGAAGGGCACACAACCCCTCAAAGGGGTGTCACTTCTT
CTGATCACTTTTGTTACTGTTTACTAACTGATCCTATGAATCACTGTGTCTTCTCAGAGG
CCGTGAACCACGTCTGCAAT
```

Figure 3.2 – Entries in a FASTA file [93]. Description headers are lines starting with the symbol ">". Nucleotide sequence follow after the header line.

Extending each entry in Fig 3.2 with a quality score for each base upgrades the format to FASTQ, a format widely used to store high-throughput sequence data 3.3.

```
@DJB775P1:248:D0MDGACXX:7:1202:12362:49613 ❶
TGCTTACTCTGCGTTGATACCACTGCTTAGATCGGAAGAGCACACGTCTGAA ❷
+ ❸
JJJJJIIJJJJJJHIHHHGHFFFFFFCEEEEEDBD?DDDDDDBDDDABDDCA ❹
@DJB775P1:248:D0MDGACXX:7:1202:12782:49716
CTCTGCGTTGATACCACTGCTTACTCTGCGTTGATACCACTGCTTAGATCGG
+
IIIIIIIIIIIIIIIHHHHHHFFFFFFEECCCCBCECCCCCCCCCCCCCCCC
```

Figure 3.3 – Elements of a FASTQ file entry [93]. (1) Description header, (2) sequence data, (3) "+" line, and (4) quality scores.

Each entry in a FASTQ file contains four elements:

1. Description header. Beginning with "@", contains the entry name or identifier
2. Sequence data. The string of nucleotides
3. The line beginning with "+". It indicates the end of sequence data, and it is optionally followed by the entry name
4. Quality scores. Sequence of numeric values reporting the confidence of each base call. A value is assigned to each nucleotide.

3.4.2 SAM/BAM format

The most common high-throughput data alignment format is the Sequence Alignment/Mapping format (SAM), and its binary analog, the BAM format. They are the de-facto standard for storing sequence reads mapped to a reference by means of an aligner.

Modern aligners output useful information about each alignment in the form of extensive amount of metadata about the sequenced samples, alignment reference, processing steps, etc. This information is included within the SAM/BAM file making their size massive. To circumvent to an extent the problem to store such large files, some research groups are switching to storing alignment data in closely related but more efficient formats like CRAM [93, 35], which is now the preferred submission format to the European Nucleotide Archive [84].

As nearly every omics pipeline involve an alignment step that produces alignment data in SAM/BAM format, a great part of bioinformatics work deals with manipulating these files. See Fig 3.4.

3.5 The case for lossy representation of the quality scores

Sequencing machines output raw sequence data, which is stored in the form of FASTQ files. When the raw sequence data is mapped onto a known reference sequence, the mapped or aligned sequences are stored in a SAM/BAM file. Due to the increasing amount of storage space

```
@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r001   99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003 2064 ref 29 17 6H5M       *  0   0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001  147 ref 37 30 9M         =  7 -39 CAGCGGCAT         * NM:i:1
```

Figure 3.4 – A sample of records in a SAM file [90]. This sample shows two lines of headers and six lines of aligned data.

required to archive the data produced by sequencing machines, the scientific community is looking into ways to improve the performance of existing genomic data compressors. In this context, the ISO/IEC MPEG working group has shown evidence of the predominant weight of QS sequences in meaningful datasets that are supposed to represent a wide spectrum of genomic data from different species and sequencing technologies [94].

Lossless genomic compressors efficiently use alignment information to store only differences with respect to a selected reference genome and the position of aligned reads, along with the length of the reads [80]. However the same strategy cannot be applied to sequences of quality scores as the concept of a reference does not exist. In addition, the wide dynamic range of QS limit their compression factor. Coding efficiency of current approaches to lossy representation of QS is also limited by the fact that there is no consensus on how far lossy compression of quality scores can be pushed, as evident in the scientific literature [45, 85, 88, 87]. Formats like CRAM include the possibility to represent quality scores in a lossy way. However, until the limitations and boundaries of lossy representation of QS become clear, and a consensus is reached in the scientific community, the opportunity of using lossy representations to achieve effective compression of sequence data, particularly at high coverage, will remain limited.

Systematic studies applying lossy compression to metadata seem to provide good hints in defining boundaries for lossy compression [42]. We believe that reaching a consensus on the adoption of lossy compression for quality scores will require an objective measure of the impact caused by the loss of information in the omics application. When comparing approaches to lossy QS representation, the two most critical aspects are:

- A clear definition of the omics application/pipeline
- Selection of appropriate metrics to measure the effect caused by lossy representation of QS such that these metrics are coherent with the context of the omics application

This is true not only for QS sequence metadata but for metadata in general where a clear understanding of their nature and use is critical to implement efficient lossy compressors with controlled impact on the host application.

3.6 Lossless compression and storage footprint

We have argumented so far the opportunity to reduce importantly the storage footprint of genomic files by focusing on reducing the size of sequence metadata, the quality scores. There has been early evidence in the scientific literature to the size occupied by sequence metadata [40, 82], and more recently such proportions have been brought back to attention [46, 81].

As a member of the ISO/IEC MPEG working group during the period of investigation of challenges in genomic information compression and storage [94], we explored footprints of sequence data files. One of the main tasks was to give an overview of the status of tools and technologies supporting genomic information compression and storage. In the light of this investigation, we added evidence to the notion of "heaviness" of the quality scores in sequence data files in both FASTQ and SAM formats with the examination of a rich collection of files spanning multiple organisms and sequencing technologies. The compiled dataset aimed, to the extent possible, at a comprehensive selection of reference sequence data. We investigated in detail this dataset, exploring the content and structure of each file. We broke down FASTQ files into components following an approach similar to [82]. Each component weighs in on the overall file size: sequence headers, nucleotides and quality scores. See Fig 3.5. Along with the file break down, different lossless compression tools were ran over each component to compare their performance.

Parallel implementations of general purpose lossless compressors like gzip and bzip2 [95, 96]

s

		file: ERR174310_1.FASTQ					
		Size (GB)	Headers (GB)	Nucleotides (GB)	Quality scores (GB)	Bits per base	Bits per QS
Uncompressed FASTQ		53.869	10.900	20.966	20.966	8.0	8.0
TOOLS							
pigz		18.597	1.570	5.982	8.639	2.282	3.296
pbzip2		14.887	1.223	5.473	7.428	2.088	2.835
scalce		10.829	1.073	3.016	6.738	1.151	2.571
dsrc		13.214	0.486	5.239	7.489	1.999	2.858
	m2	11.598	0.572	4.773	6.253	1.821	2.386
quip		11.312	0.472	4.556	6.284	1.738	2.398
	asm	11.312	0.472	4.556	6.284	1.738	2.398
fqzcomp		11.320	0.408	4.623	6.288	1.764	2.399
	extra	11.673	0.408	5.167	6.097	1.972	2.326
fastqz		10.955	0.448	4.312	6.194	1.645	2.364

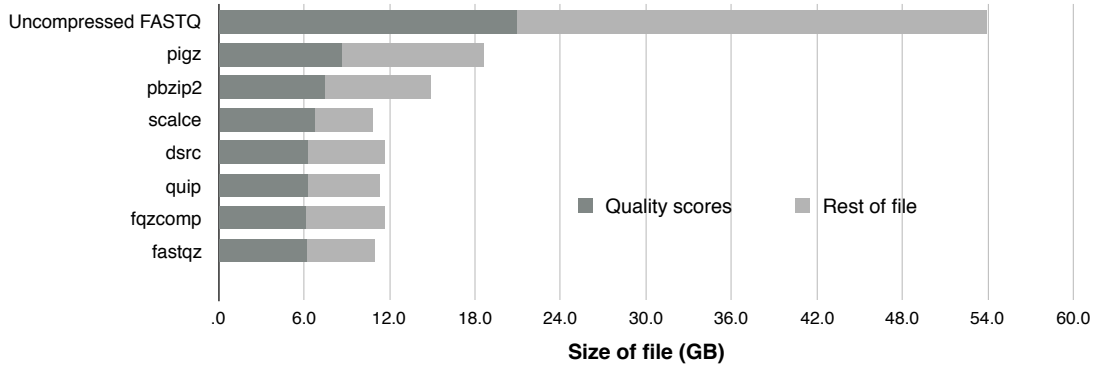


Figure 3.5 – FASTQ file for human sample. Break down of a FASTQ file in components with their corresponding storage size and lossless compression results. Compression tools: pigz [95], pbzip2 [96], scalce [41], dsrc [97], quip [40], fqzcomp and fastqz [82].

and specialized lossless compressors for sequence data [41, 97, 40, 82] turned out the same result for the quality scores. For all intents and purposes, such metadata poses a heavy weight on sequence data files. Refer to the bottom graph in Fig 3.5.

The quality score metadata component in FASTQ files can be contrasted with the proportion of file size it utilizes. The amount of storage needed for quality scores is overwhelmingly large for data that is meant to support the actual genomic information comprised of nucleotide sequences. Refer to figures 3.6 to 3.8.

Lossless ad hoc compressors for sequence data in FASTQ files give definite evidence of storage savings. See top graphs in figures 3.6 to 3.8. The limit to sequence data compression however

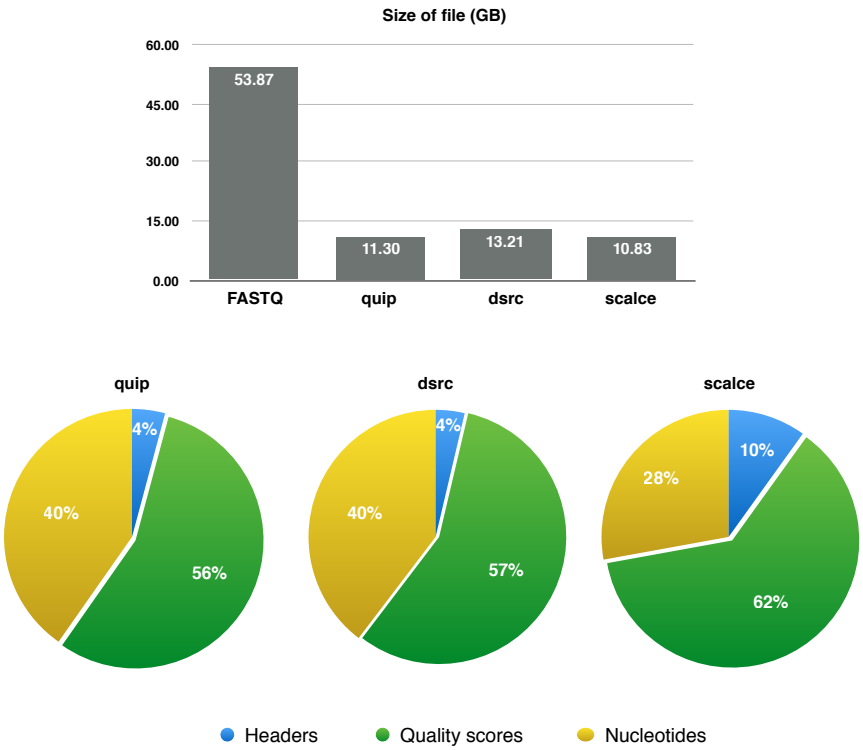


Figure 3.6 – FASTQ file for human sample and its break down. Dataset file ERR174310_1.

is currently being set by the quality scores. See pie charts in figures 3.6 to 3.8.

Our observation is that no less than 50% of the file size of sequence data is dedicated to store quality scores. Notwithstanding the nature of the data, whatever the sequenced organism or species. We confirmed this observation for every file in the dataset under test, supporting the claim that the DNA sequence portion accounts for a minority of the disk space, yet is the primary purpose for the file [82].

3.7 Bioinformatic workflows and pipelines

Bioinformatic analyses involve shepherding files through a sequence of transformations, called a workflow or pipeline [98]. We can outline a bioinformatic pipeline in four main steps from source to end: acquisition, preprocessing, processing and analysis. An acquisition step draws in sequence data from a centralized repository. The quality of data is evaluated with the

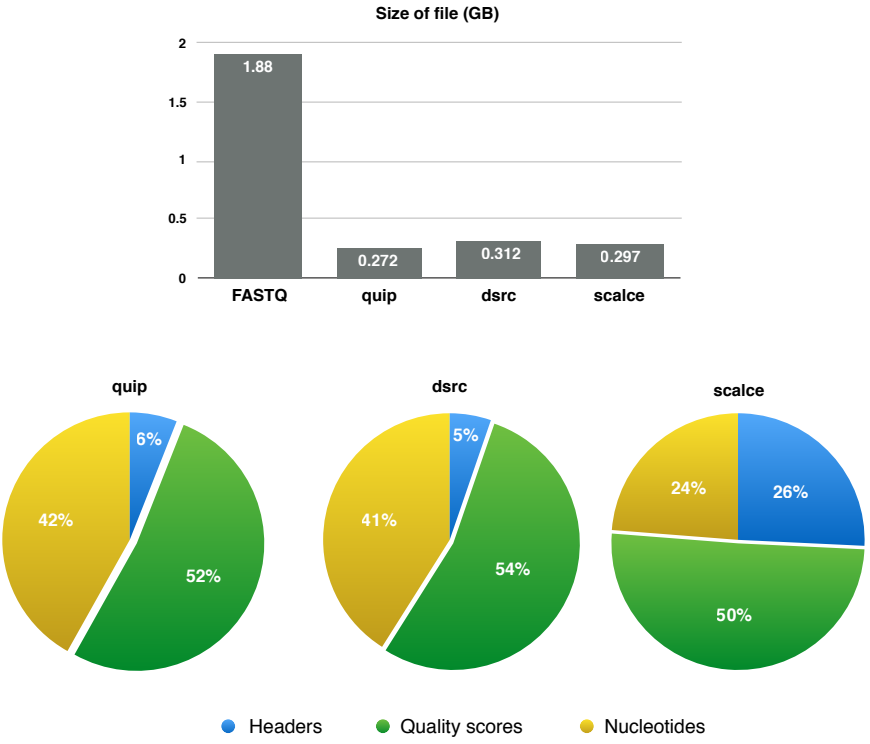


Figure 3.7 – FASTQ file for metagenomics (human gut) and its break down. Dataset file MH0001_081026_clean_1.

purpose of cleaning it and correcting it through preprocessing. Following this step, sequence data is ready to be sourced to relevant omics applications of interest, which are themselves computational pipelines, and it is where the actual computation takes place. The final step generally consist of a summary reporting results.

The challenges associated with the implementation of bioinformatics pipelines are well documented. As per [28], they revolve around analysis provenance, data management of massive datasets, ease of use of computational tools and interpretability and reproducibility of results. With the increasing complexity of bioinformatic analyses it is becoming more common to rely on frameworks to help process sequence data and metadata. Platforms such as Galaxy [99] and others [98] aim to address limitations in current data-driven biomedical science. The final goal is to make bioinformatic analyses more accessible to all researches, ensure reproducibility of results and facilitate their communication.

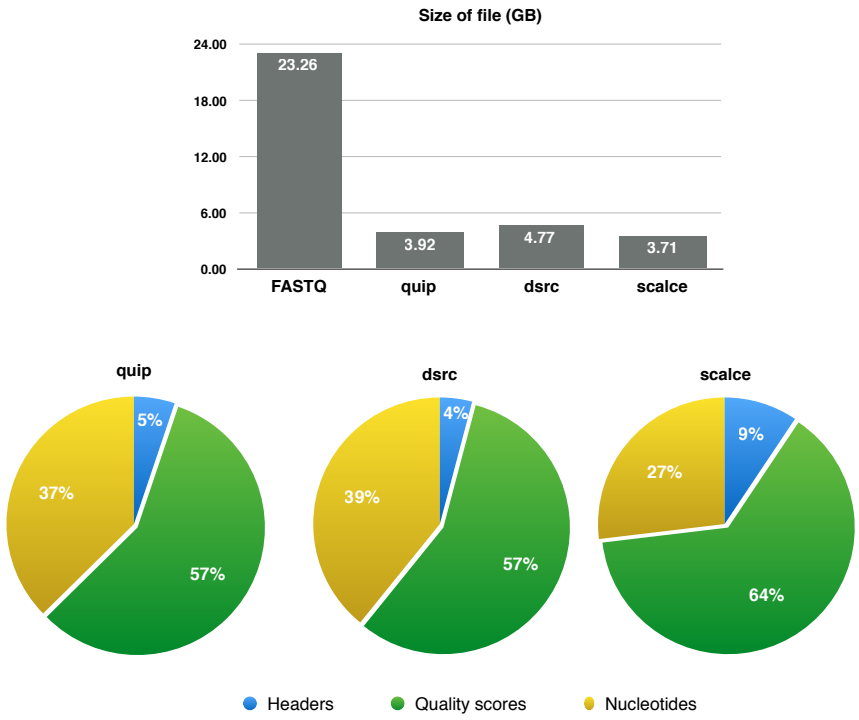


Figure 3.8 – FASTQ file for cacao plant and its break down. Dataset file SRR870667_1

Multiple research groups around the world are developing omics applications, most of which have arguably appeared in the last decade. Their statistical and algorithmic approaches are intricate and very sophisticated. Further, the accuracy of the methods is constantly updated. These efforts of improvement lead to inevitable changes in the stability of pipelines that can be put together at any given time, and add to the large repertoire of computational tools available to try out. A quick look into perhaps the largest catalogue of omics tools [50] turns out more than 30,000 indexed software tools, spanning applications in genomics, epigenomics, transcriptomics, proteomics, metabolomics and phenomics.

Each omics application is *sui generis*. It leverages on particular protocols of a sequencing method [100] and a unique set and sequence of logic must be set in place, which requires the selection of appropriate software and algorithms to be assembled for execution in the pipeline. Building a bioinformatic pipeline generally consist of a mix of open source tools alongside custom scripts.

3.8 Testbed for impact analysis

We move on now to describe the organization of the testbed we set up to investigate the impact of lossy representation of quality scores. We laid out our platform for testing in accordance to the multistep organization of a bioinformatic pipeline. The acquisition, preprocessing, processing and analysis steps are shown in Fig 3.9.

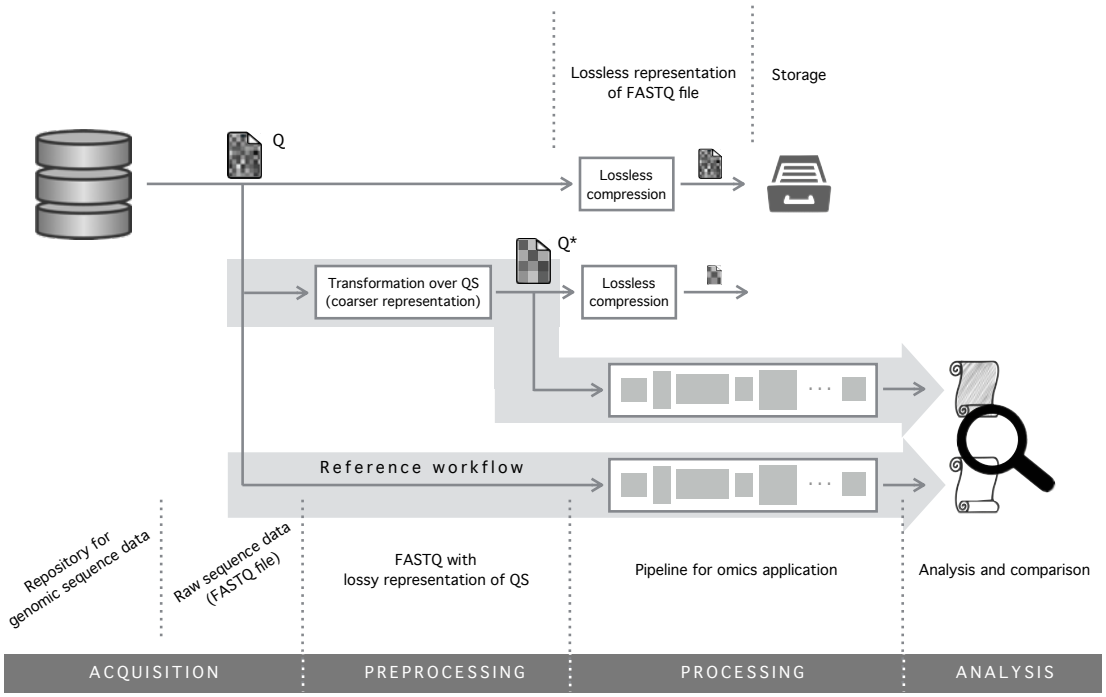


Figure 3.9 – Testbed for evaluating the impact of lossy representation of QS. The organization follows the multistep structure of a bioinformatic pipeline.

The testbed workflow is as follows:

1. Acquisition. Data retrieval from a centralized repository. There is a sheer number of available online databases to the extent that summaries of updated sources are published on a yearly basis [101]. For the most part we relied on the National Institute of Health (NIH) genetic sequence database[1], GenBank [102], and on the European Bioinformatics Institute (EMBL-EBI) database[2] to source sequence data. In addition,

[1] https://www.ncbi.nlm.nih.gov/
[2] https://www.ebi.ac.uk/

we also used the dataset in [94].

2. Preprocessing. As our interest lies in the quality score metadata of sequence data files, we assigned this step to their operation. Approaches to lossy representation of QS differ by the transformation implemented in the algorithm for lossy representation. With the intent to reduce storage footprint, these algorithms output a new and more "compressible" QS representation. Refer to the top of Fig 3.9.

   Few lossy compression methods for the quality scores integrate encoding for the new QS representation [88]. Most do not however. The treatment of QS is generally different for every lossy compressor and wether they require the QS to be sourced from FASTQ or SAM formats, compressors pursue the same goal: to reduce the entropy of quality score metadata.

   The preprocessing step contains all computations done over quality scores keeping the nucleotide sequences intact. After preprocessing, a FASTQ file with a new representation for the QS is output.

3. Processing. The core bioinformatic treatment lies in this step. There is a clear trend in bioinformatics toward the adoption of workflow tools for automation and creation of research pipelines [103]. Currently however, bioinformatic analyses remain to a large extent file-based without standardization of data flow in the workflow [98].

   Pipelines for omics applications are themselves bioinformatic pipelines. They deploy the same outline, sequentially piling software tools on input data. A core element in the pipeline is sequence alignment, which we will discuss later and in detail in the following chapters.

4. Analysis. After a successful run of the pipeline, relevant results are collected and inspected. This is the final step in the testbed workflow.

To test the impact of lossy QS representation we make the reasonable assumption that two identically built and configured omics pipelines will output the same result, provided that we input the same data entry. However, that in principle could be argued [47, 55].

On the testbed we run independently two workflows that are identical in every way but whose inputs are different. Refer to Fig 3.9. The reference input is a FASTQ file without any corruption of their quality scores, and is the data entry to the reference workflow.

For the second workflow we source a different data entry to the omics application. We deliberately change beforehand the data representation of QS by means of a lossy compressor and reformat the original FASTQ with this new representation. We follow by feeding the omics pipelines with the corresponding FASTQ file input and obtain the result of both workflows. The outputs are then collected and compared in the analysis step of the testbed.

We followed the above procedure for the omics application variant calling. In the next section we overview the pipeline behind this omics application.

3.9 Bioinformatic pipeline for variant calling

A variation at a single position in the genome among individuals is called a single nucleotide variation (SNV). Some variations are expected at a given genomic locus, and are found in the population at an arbitrary low frequency. For example, if more than 1% of the population does not carry the same nucleotide at a specific position in the DNA sequence, the nucleotide variation is called single nucleotide polymorphism (SNP).

We will use the name variant calling for the process of identifying genetic variants, and abuse the term to refer indistinctively to the identification of both SNVs or SNPs. We say then that identifying nucleotide variations in the genome is the goal of the omics application variant calling.

The pipeline for variant calling can be organized in six steps as shown in Fig 3.10. The sequence of steps starts with the preprocessing section, which is shared among most bioinformatic pipelines. The processing section contains the steps specific to variant calling.

The workflow for discovering variants is as follows:

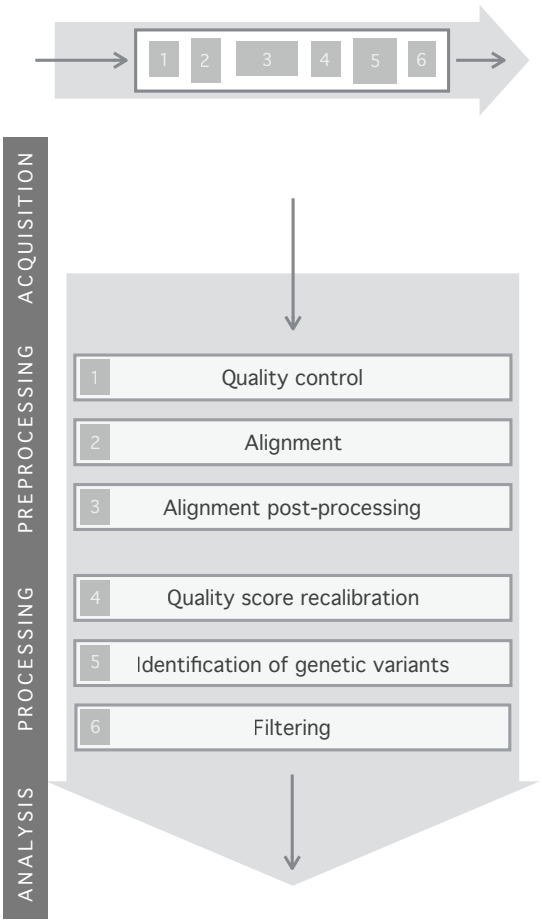


Figure 3.10 – Processing pipeline for variant calling. Outline of the main steps.

1. Quality control. Base calling procedures vary among sequencing platforms and all of them present errors. Depending on the platform, overall error rates range from 0.1% to 13% [63]. Reducing error rates of base calls and improving the accuracy of per-base quality score impact importantly the assembly and alignment of sequence reads, as well as the detection of genomic variants.

   Inspecting the quality of genomic sequences consists, among other things, in checking the distribution of quality scores at each sequence position, check for over represented sequences and look for deviations from the expected nucleotide content.

   There are well know artifacts typical to all sequencing platforms that can be simply overcome by read trimming. Meanwhile some sequencing platforms discard altogether

poor quality reads, so as to prevent hindering sequence alignment [104].

2. Alignment. The accuracy of sequence alignment has a crucial role in variant discovery. Reads that are wrongly aligned may result in artificial deviations from the reference, leading to errors in variant detection. When mapping sequences to a reference genome it is important that aligners, whenever possible, cope well with both sequencing errors and actual biological differences due to polymorphisms. Furthermore, it is essential to set a mapping criterion to limit the allowed amount of sequence identity between each read and the reference. Also, the tolerable number of mismatches may vary between different organisms, or if a mismatched reference is used, so this is another choice to be made.

   Alignment of reads to repetitive regions in the reference genome is a well known challenge for sequence alignment [105]. Alignment is also more difficult for regions with higher levels of diversity between the reference and the sequenced genome [106].

   We note that comparative analyses studying the impact of read alignment algorithms to final variant call sets in combination with multiple variant calling methods [59] point out the critical influence of both aligners and variant callers in the discovery of variants [59, 104]. Other studies have found that sequence alignment can play as vital a role in variant detection as variant callers themselves [56].

3. Alignment post-processing. This is the last preprocessing step in the pipeline. It prepares the aligned reads for the variant caller by sorting them, removing duplicated alignments and realigning reads around difficult genomic regions. Reads are organized and usually sorted by chromosomal positions to facilitate future search. Non-unique alignments introduce ambiguity in the calling, as reads aligned to multiple positions in the reference genome are commonly considered indistinctive, hence duplicated alignments are identified and removed.

   Aiming at lessen to some extent the identification of artifactual variations due to uncertainties in alignment, reads aligned to problematic regions are “resolved” by aligning them again. Local realignment identifies the most parsimonious alignment along all

reads at a problematic locus by finding a consensus sequence, whose selection relies on a score based on the quality scores [107]. Realignment around target regions helps improve the accuracy of the downstream processing steps.

4. Quality score recalibration. In this step systematic errors made by the sequencer when it estimates the quality score of each base call are corrected. In this process, patterns in how these errors correlate with nucleotides are identified and a model of covariation is build. The model is based on the data and a set of known variants. The covariation is analyzed among several features of a base: reported quality score, the position of the base in the read, and the sequencing context of the base (preceding and current nucleotide) [108]. The model applies corrections to adjust the quality scores of all reads in the input file, recalibrating their values.

5. Identification of genetic variants. Earlier approaches counted the abundance of observed alleles (variant form of genes) at every site. Then, filters with fixed cutoffs based on quality scores were applied to keep only high-confidence bases from which variants were called. More powerful methods have been developed that integrate several sources of information within a probabilistic framework, and provide a natural way for quantifying uncertainty about the variant call [104]. These methods leverage on the quality scores for each read to calculate posterior probabilities to determine genotype likelihoods [106].

   Multiple variant callers are available but their low concordance is a problem for accurate and consistent identification of genomic variants. Several studies have evaluated their effectiveness [53, 109] and have also done it in combination with different alignment strategies [56, 110, 111, 59]. In the literature we have found general recommendations for the selection and configuration of variant callers. The choice largely depends on the type of variant of interest and the available data [43]. It is widely acknowledged that there is no one-size-fits-all method. What is constant however is the advise to exercise caution when analyzing results, as well as in the interpretation of positive and negative findings [110].

6. Filtering. Variant callers usually come with their set of specific filters with recommendations thereof. Filtering attempts to reduce the amount of false positive calls, improving the calling. While it might remove authentic variants, it also minimizes artifacts overall. Artifacts stem from the process of preparing the biological sample for sequencing, sequencing itself and from alignment. Traditional model-based variant callers rely heavily on ad-hoc filters because artifacts are produced in very complex ways that are beyond simple modeling. As a result, variant callers need to be fine-tuned to achieve the expected accuracy on naive datasets. Yet, their optimal parameter values are unknown to the tester, and some of them can only be understood or safely tuned by the developers [91, 95].

   Although the application of caller-specific filters complicate comparison and obscure artifacts, it is possible to define a set universal filters applicable to most callers and compare their effect for different variant calling configurations [112].

The process of calling can result in thousands of variants. For example, the pipelines presented in [104], ran on whole human exome datasets, can generate about 24,000 variants.

Discovered variants are annotated and compared against a truth set (also “gold standard” or "golden reference") to check the performance of the analysis pipeline. The evaluation of variants, or concordance verification, is commonly expressed as the percentage of variants in the sample that match (are concordant with) variants in the truth set. See Fig 3.11.

3.10 Framework for the evaluation of lossy quality scores in variant calling

Building on evaluation strategies put forward at the time in the state-of-the-art [41, 37, 86, 88], we provide the specification and initial validation of an evaluation framework for the comparison of lossy compressors for genome sequence metadata. This work was spurred by the ISO/IEC SC29/WG11 technical committee (MPEG) at the onset of the standardization activity

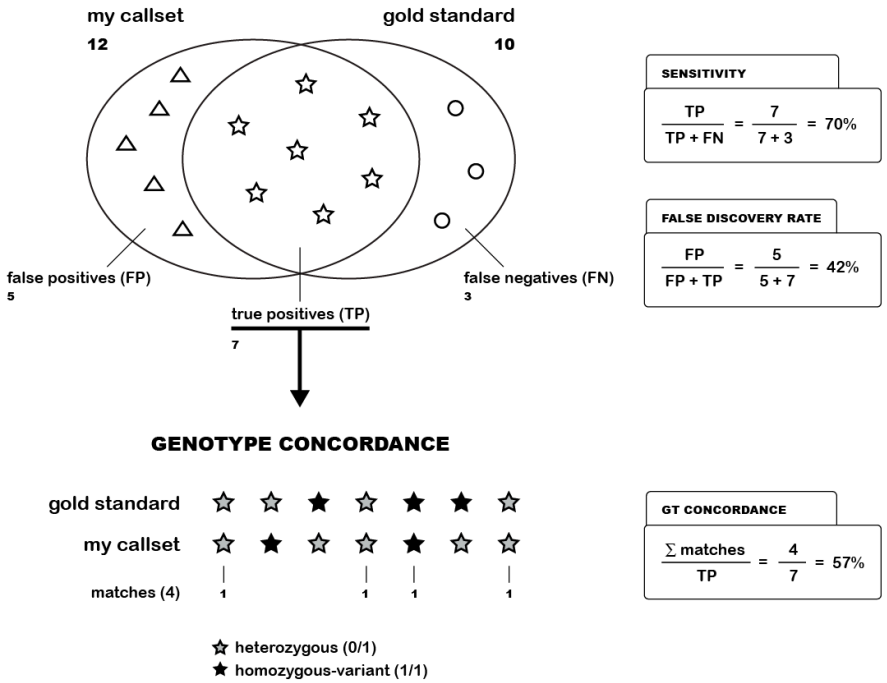


Figure 3.11 – Variant evaluation and concordance verification. Image borrowed from [113].

for genomic information representation. It was developed in collaboration with Stanford University, the Massachusetts Institute of Technology and Leibniz University Hannover, and published in [114].

The goal of the framework is to define reference data, test sets, tools and metrics that can be used to evaluate the impact of lossy compression of quality scores on human genome variant calling. The functionality of the framework is validated referring to two state-of-the-art lossy compressors for the quality scores.

The methodology to evaluate the effect of lossy QS metadata requires the identification and definition of the following elements:

- The types of data to be analyzed
- The genomic analysis applications addressed
- The specific data to be analyzed (both test data sets and golden references)

- The analysis tools used to perform the analysis

- The metrics to evaluate the "inaccuracy" or "errors" induced in the calling of variants

3.10.1 Type of sequence data: Human genome

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

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

Because of the large impact of the mentioned studies, we addresses only variant calling of the human genome. Future work should extend this to satisfy other analysis applications and species, primarily in three areas. The first of these is metagenomics, the study of genetic material extracted from environmental samples. Because the microbial community contained in the gut plays an important role in protecting against pathogenic microbes, modulating immunity and regulating metabolic processes, the human gut microbiome is of significant interest to human health. The second area for future work is variant calling in cancer genomes; mutations discovered in genetic material extracted from tumor cells can play an important role in oncology with the possibility to define targeted and personalized therapies. Finally, future work should extend to other species, which include infectious disease agents whose

genetic signature can be crucial for the derivation of sequence-based markers of pathogen identity, antimicrobial resistance, virulence and pathogenicity to advance therapeutic decision making systems.

3.10.2 Datasets and human reference genome

The dataset necessary to perform a variant calling analysis is composed of: a reference genome used to identify and catalog mismatches; several samples generated from the same sequenced individual using different sequencing technologies and different configuration of the sequencing machines; and high-confidence variant calls generated by several orthogonal experiments and considered of high quality by the scientific community. Within such data sets, high-confidence regions are usually identified and separated from lower quality variant calling results.

Reference genome

Even though the human reference genome GRCh38 has already been published by NCBI, the largest part of available sequence data and the related variants calls have been produced with previous publications of the reference. GRCh38 also has alternative "contigs" (set of overlapping DNA segments that together reconstruct a larger DNA sequence) and most current methods have not been adapted to work well with this new reference. Therefore, the selected reference human genome to be used is a previous version, the assembly GRCh37[3].

Sequence data and gold standard

This work is considering individual NA12878 as published by the Coriell Cell Repository [115]. This individual is part of a trio (parents and son) that has become a reference in literature and it is currently part of two initiatives, the Illumina Platinum Genome project [116] and the Genome in a Bottle (GIAB) initiative for the definition of high confidence genomic variants

[3] http://www.ncbi.nlm.nih.gov/assembly/2758/

Table 3.1 – Dataset for individual NA12878

ID	Description	Source
1	NA12878 from IonTorrent	SRX517292 [117]
2	NA12878 replicate J – 8bin QS, 30x Illumina 8-binned QS	Garvan [118]
3	High coverage Illumina dataset with non-binned QS	SMaSH dataset (Berkeley) (50x) [119]
4	Run SRR1231836 of experiment accession SRX514833 stored on the DDBJ repository	SRX514833 [120]

calls data. The sequence data selected is listed in table 3.1.

Illumina and IonTorrent sequencing technologies have been selected as they represent the largest share of the sequencing machines in use and most of the data stored in public repositories were produced using these technologies. Illumina samples include 8-binned QS, which is currently the default configuration for the latest Illumina sequencing machines. This indicates that the common usage is already exhibiting a partial loss of the original machine-generated accuracy for QS. The dataset in table 3.1 should be updated in the future to consider new generations of sequencing machines that might have different behaviors and performance when producing QS.

The gold standard variants for individual NA12878 considered for this study were two:

- The Illumina Platinum Genomes High confidence variant calls[4]
- The GIAB-NIST reference variants[5]

Variant calling tools and metrics

The core tools in the pipeline for variant calling are the sequence aligner and the variant caller. Four pipeline configurations were explored for this study and are listed in table 3.2. Detailed specification of parameter settings and values are reported in [114, 79].

[4] http://www.illumina.com/platinumgenomes/
[5] ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/NA12878_HG001/NISTv2.19/

Table 3.2 – Pipelines for variant calling considered for evaluation.

ID	Aligner	Variant caller	Pipeline
1	BWA-MEM [121]	GATK_HC [122]	BWA-MEM+GATK_HC
2	Bowtie2 [123]	GATK_HC	Bowtie2+GATK_HC
3	BWA-MEM	SAMtools+BCFtools [83]	BWA-MEM+SAMtools+BCFtools
4	Bowtie2	SAMtools+BCFtools	Bowtie2+SAMtools+BCFtools

Table 3.3 – Variant calling performance metrics. TP= true positive, TN= true negative, FP= false positive, and FN= false negative. Table adapted from [37].

Metric	Synonym	Formula	Relation with other metrics
Sensitivity	Recall, True positive rate (TPR)	$\frac{TP}{TP+FN}$	
Specificity		$\frac{TN}{TN+FP}$	
False positive rate (FPR)		$\frac{FP}{TN+FP}$	1-Specificity
Positive predictive value (PPV)	Precision	$\frac{TP}{TP+FP}$	
False discovery rate (FDR)		$\frac{FP}{TP+FP}$	1-PPV
F-score	F_1 score	$2 \times \frac{Sensitivity \times PPV}{Sensitivity+PPV}$	Harmonic mean

For variant callers, commonly used performance metrics include sensitivity, specificity, false positive rate, positive predictive value (PPV), false discovery rate (FDR), and F-score [37, 124]. The definitions are shown in table 3.3.

To asses the correctness of the calling we use the metrics Sensitivity, Precision and F-score, such that:

- TP is the number of variants in the gold standard that have been called and marked as positive by the variant caller
- FN is the number of variants in the gold standard that have not been called or have been called, but marked as negative by the variant caller
- FP is the number of positions that have been called and marked as positive by the variant

caller, but are not in the gold standard

The F-score provides a way to balance the effects of sensitivity and precision and will be used as an additional measure. It ranges from 0 (worst score) to 1 (perfect score).

The ROC curve is also a very commonly used metric in benchmarking studies to visually illustrate the trade-off between sensitivity and specificity. It is defined as the plot of False positive rate versus the True positive rate. The area under the ROC curve (AUC), a fraction between 0 and 1, measures the overall accuracy under a range of variant calling thresholds.

In the literature we have largely found sensitivity and precision as prevalent metrics for measuring the performance of pipelines for varant calling. In our experience, the use of AUC is not readily seen for variant evaluation. In fact, it has been reported that AUC should only serve as a supplementary metric because it does not inform the accuracy under optimal or default threshold [37]. However, graphical presentation of performance metrics can facilitate the comparison and evaluation of algorithm performance [124].

We note that different methods for calling variants output different sets of calls. Some methods privilege the generation of small output files containing mostly TPs, while others generate larger outputs with larger amounts of both TPs and FPs; evaluating their concordance is an open probem, as noted in section 3.9. Nevertheless the selection of metrics presented in this section can, for all intents and purposes, quantify the effect of lossy quality score representation in the calling of variants.

3.10.3 Comparison of tools

To validate the proposed approach we compared the variant calling results obtained with lossy compression of QS using QVZ [85] and Quartz [87] on the dataset with ID 4 listed in table 3.1. The golden reference for variant calling was the Illumina Platinum Genomes v8.0. The tests were run for the four pipelines listed in table 3.2 and organized as shown in Fig 3.12. The computational infrastructure was an Intel Xeon CPU E5-2660 v3 at 2.60GHz with 251 GB RAM,

running CentOS Linux release 7.1.1503. Results are shown in Fig 3.13.

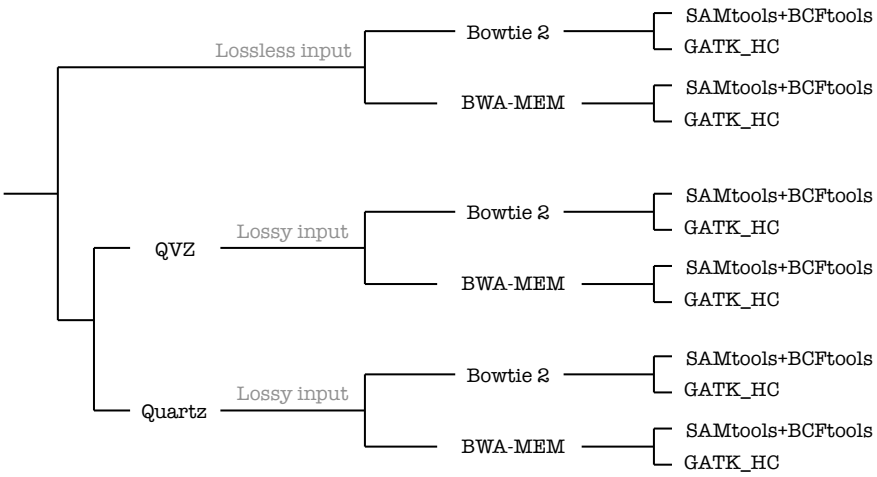


Figure 3.12 – Organization of pipelines tested in the framework.

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

The results shown in Fig 3.13 are reported here as first validation of the proposed approach. The figures indicate that the implementation of lossy compression of QVZ has a smaller impact on variant calling than the one of Quartz when using BWA-MEM for alignment, though, in this case, Quartz is using less than half the bits per QV compared to QVZ; a tradeoff between compression and accuracy certainly exists.

When using Bowtie2 as aligner, both Quartz and QVZ show better precision with respect to the lossless case. Lossy compression has a higher impact on the pipeline using Bowtie2 because in this case QS are used in the alignment process. A more detailed inspection should be conducted to explain the drop in precision and sensitivity in the pipelines featuring BWA-MEM.

Further validation of these results will require enlarging the experiments to the whole dataset proposed here. We note that in [87] an improvement in AUC with respect to the lossless case was reported; this is a different measure of performance from what we present here. Running times in figure 3.13 are provided for completeness.

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

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

Figure 3.13 – Impact of lossy compression of QS on variant calling for the configurations listed in table 3.2 and depicted in Fig 3.12.

3.11 Discussion

In this work we defined a framework to measure the impact of lossy quality scores on variant calling for human genomes. This framework defines test sets, reference data, and tools to perform variant calling together with their processing configurations. A precise definition of testing conditions is of utmost importance to enable reproducibility of results, as well as comparison and ranking of the compression tools under evaluation.

It is important to remark that the main goal of the methodological framework is to assess the effects of lossy QS compression on variant calling with respect to the reference lossless compression case, and not to understand if better or different variant calling results are obtained when applying lossy compression. Although some works have suggested that QS

are affected by noise [87], and that it might be possibly filtered out by a lossy compression stage, here we abstain from any considerations regarding the actual quality of analysis results obtained.

An immediate improvement to the quantification and understanding of the effect of lossy QS representation in variant calling will come from the detailed examination of its pipeline. The collective effect of every step in the processing pipeline is reduced to a couple of values that reflect the performance of the calling. Clearly, the effect produced by a lossy representation of QS cannot be isolated to be analyzed independently, but it may serve well to focus instead on their progressive transformation along the pipeline.

It could also be useful to find a consensus on the selection of methods, like for the case of filters, as they discriminate harshly, deciding the quality variants that make the final call (refer to section 3.9). In [112] the application of caller-oblivious filters to derive the final call set is suggested to improve variant accuracy between distinct pipelines.

Arguably, with summarization metrics like sensitivity or precision, we are limited to interpret results at a granularity level that may be insufficient for an application that looks for tiny changes in the genome (nucleotides differences in less than 5% of the reference genome). What is more, in our tests we found the variation of these metrics to be in the order of 0.1%. More recent studies have shown configurations for variant calling with lossy QS, reporting even smaller order of variations for precision between 0.01% [46] and 0.001% [45]. These figures are claimed immaterial and they may very well be. However, it could be the case that the metrics may not be adequate to reflect the small differences represented in variants. Further, it might be that manual inspection of each relevant call is necessary.

Early on it was reported that most discrepancies in the calling of variants came from marginal decisions between homozygote (two identical alleles at a particular genomic locus) and heterozygote (two different alleles at a particular genomic locus) calls [37]. Also, for all discrepant cases the read coverage was very small in comparison to the mean read coverage. Further, it was reported that lossy representation of QS pushed allele quality marginally over the

threshold to be called a quality variant.

Consequently, it is very likely that the effect of lossy QS representation cannot be discerned from other artefactual events in the pipeline, and we strongly suspect this to be the case.

4 Lossy quality scores and differential gene expression

The genome of the cell is the total of its genetic information as embodied in its complete double-stranded DNA molecule. The genetic information is encoded through the order of the nucleotides along each strand of the DNA sequence. The order of the nucleotides spells out biological messages. In order to put the genetic information stored in the DNA into action the biological messages must "express", which guides the synthesis of other molecules in the cell. This flow of genetic information is a mechanism shared by all living organisms, and leads to the production of RNA molecules and protein molecules. RNA molecules are working copies of the information stored in sets of given segments of the DNA sequence, and are used as templates to direct the synthesis of proteins. Protein molecules are the principal catalysts for almost all chemical reactions in the cell, they are building blocks, and perform particular functions depending on their own amino acid sequence. Each sequence of amino acids is specified by the gene that codes for that protein, that is, it is specified by the nucleotide sequence of corresponding set of segments of DNA.

Genes can express through their nucleotide sequence the genetic information they store. Gene expression is the process through which a cell converts the nucleotide sequence of a gene, first into the nucleotide sequence of an RNA molecule, and then into the amino acid sequence of a protein [125]. The expression of genes, and cellular processes in general, are complicated to understand primarily because of the degradation and transience of molecules within the

cell [126]. However, sample preparation enable snapshots of cellular metabolism and activity to be captured with high-throughput sequencing. With RNA sequencing (RNA-seq) we can get information about the content of RNA in a sample, and its abundance can be used as a proxy to measure the expression level of genes [127]. Further, genes expression levels can be measured between samples to identify differences in their expression profiles. This application is called differential gene expression.

In this chapter we evaluate the use of representing lossily the quality scores in the omics application differential gene expression. We start with the fundamental context focused on genes and transcription, and describe how RNA-seq enables the extraction of information of RNA content in a sample. Then we explain the core processes in the measure of gene expression and comment on their problems. We continue with the organization of the proposed pipeline, and present a strategy for evaluating the impact of lossy quality score representation in the calling of differentially expressed genes. In the final part we present results and a discussion.

4.1 Organization of genetic information: chromosomes and genes

When a cell needs to read out its genetic instructions it scans the relevant sequence of nucleotides contained in the DNA and copy that portion into RNA; this portion comes from a region called gene. The segment of DNA sequence that is copied comes from a region called gene. In eucaryotes organisms like the human, where the DNA resides in the cell nucleus, the DNA is divided between a set of different chromosomes, which further divides into genes. How the genome is organized into chromosomes depends on the eukaryotic species. Thus, the genes, the functional units of heredity, are carried along by the chromosomes in which the DNA organizes. Each chromosome consists of a single, very large DNA molecule. In humans, each cell contains 46 chromosomes, organized in 22 pairs plus two sex chromosomes. With the exception of the germ cells and a few specialized cell types that cannot multiply and lack DNA altogether [125], each human cell holds two copies of each chromosome, which result from homologous recombination.

Chromosomes carry genes but they also contain a large excess of interspersed DNA nucleotides that do not seem to carry critical information. The size of genomes vary widely, primarily because of differences in the amount of DNA scattered between genes. Although the utility of the interspersed DNA has yet to be demonstrated, some of it is crucial for the proper expression of genes.

In Figure 4.1 the organization of genes on a human chromosome is shown. The image was borrowed from [125]. At the top of the figure, the schematic of a human homolog chromosome pair composed of two DNA molecules is shown. The chromosome depicted is one the smallest human chromosomes, chromosome number 22, wich makes up approximately 1.5% of the entire human genome. In Figure 4.1(B) a portion of the chromosome is expanded into a drawing that highlights a sequence of colored vertical bands, each of which represent a gene. In this expansion, 40 genes are indicated. The spatial organization of genes and the interspersed DNA between them, the intergenic regions, is shown up close by zooming in into a segment of the previously expanded portion of the chromosome. Refer to Figure 4.1(C). Lastly, an arrangement of the fundamental components of genes, namely exons, introns and regulatory sequences, is shown for a single gene in Figure 4.1(D).

Most of the DNA in a gene consist of long stretches of nucleotides that interrupt the rather small regions of DNA that code for proteins. The coding sequences in a gene are called exons and the intervening, non-coding sequences, are called introns. In addition, each gene has associated a regulatory sequence of nucleotides, which determine the correct expression of the gene, at the right time and in the right type of cell [125].

4.2 Gene expression and transcription: from DNA to RNA

Each cell in an organism access their genetic information by first reading the genes in their DNA. Then, ”copies“ of nucleotide sequences of genes are produced and represented into a different chain of nucleic acids, in a process called transcription. Typically, a cell expresses only a fraction of its genes, which means that only certain genes will be read out and transcribed

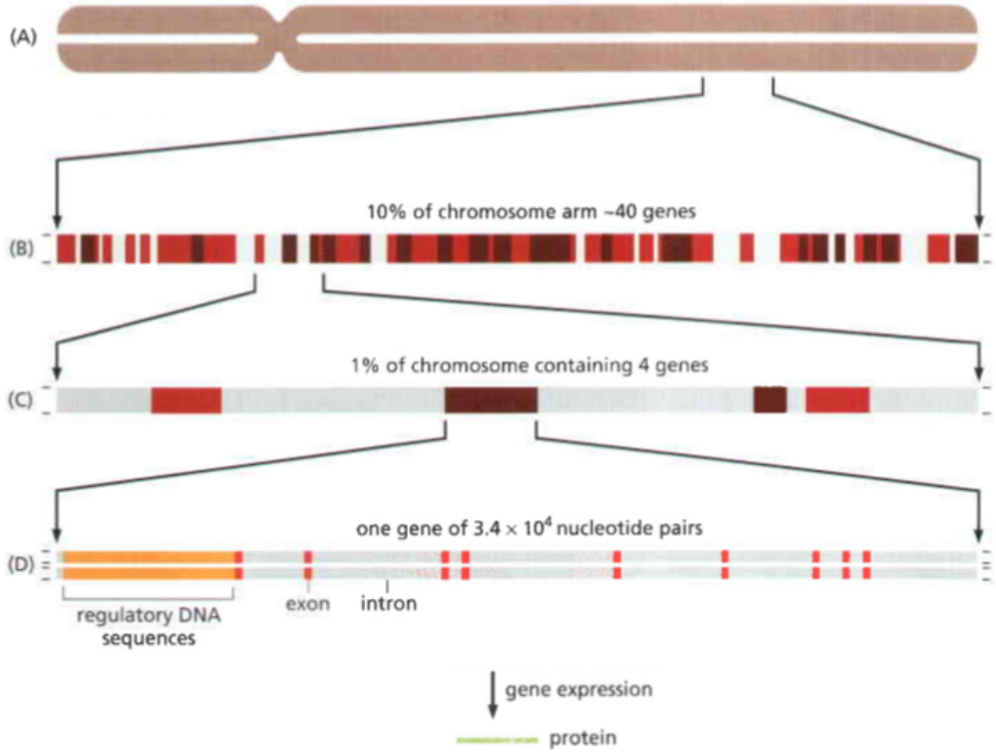


Figure 4.1 – Spatial organization in a human chromosome [128]. (A) Stylized representation of human chromosome 22; (B) Band pattern depicting the distribution of genes in a segment of a chromosome; (C) Chromosomal region holding four genes; (D) General organization of elements within a gene.

at any given time. When a gene is expressed the corresponding nucleotide sequence of DNA is transcribed into a separate, single-stranded molecule of RNA, and it is said to be transcribed into a RNA nucleotide sequence. Thus, in the process of transcription, nucleotide sequences of DNA are transformed into chains of RNA; refer to the left side of Figure 4.2. Each transcribed segment is called a transcription unit, and a transcript is the RNA chain produced by transcription.

The production of RNA is controlled by the cell in such a way that the amount of transcripts produced from the same gene is regulated, as it is also the translation of transcript information into proteins. In the example shown in Figure 2, genes A and B are expressed differently, each produces a different amount of transcript abundance, and the translation of their respective proteins, protein A and B, is carried out at different rates.

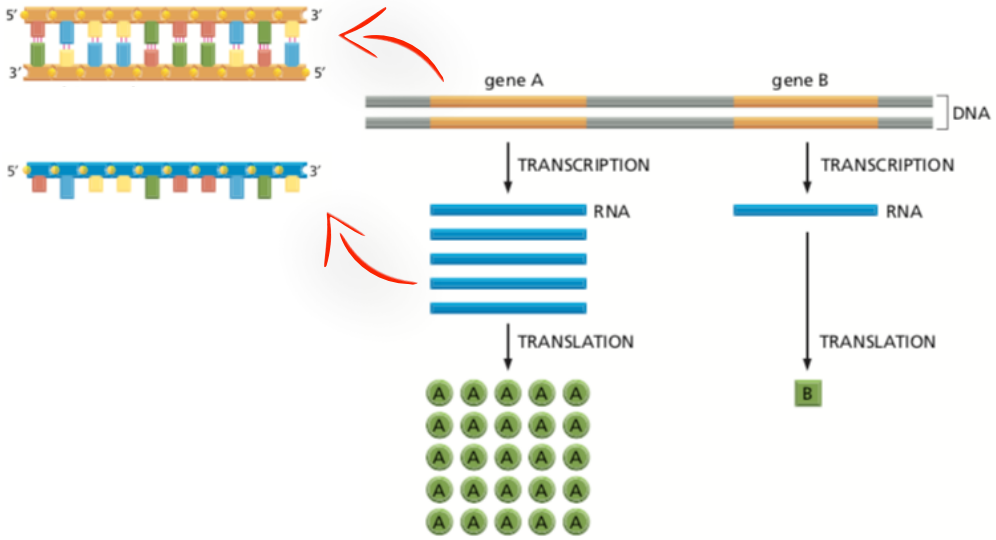


Figure 4.2 – Expression of genes. Through the process of gene expression DNA is copied into RNA and translated to proteins. RNA plays primarily an intermediary role in the synthesis of proteins. Figure adapted from [125].

4.2.1 Alternative splicing

All eukaryotic cells express their genetic information in the same way, via the pathway from DNA to protein. This principle is called the central dogma of molecular biology. The RNA transcripts are intermediaries in the transfer of genetic information, acting as messengers that make possible the synthesis of proteins. Many types of RNA molecules are produced within the cell and the type that codes for proteins is called messenger RNA (mRNA). Often, RNA molecules transcribed from the same gene are processed differently, giving rise to the production of variations of the same transcribed sequence. Each of these variations spells out different information; a consequence of differences in the organization of transcribed coding regions (exons). The process of RNA splicing alludes to the connection of exonic regions and the removal of intervening intron sequences before translation. Different organization of exons produce distinct mRNA sequences, which in turn translate to different proteins. The mRNA sequences coming from these alternative representations of a gene are called isoforms of the gene, and their exons are said to be alternatively spliced. Three alternative representations, or isoforms, of the same gene are shown in Figure 4.3. The arrangement of

exons in each isoform is different and so is the number of sequences expressed per isoform, that is, their abundance.

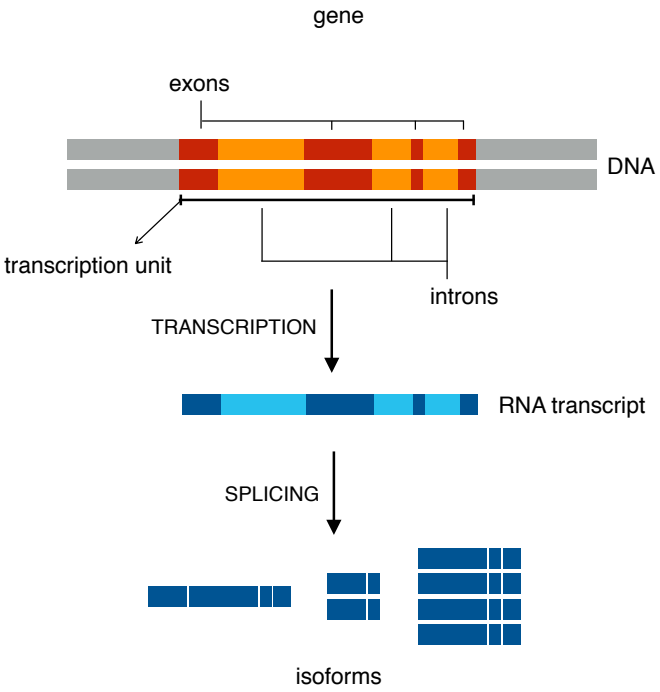


Figure 4.3 – RNA splicing. Before the RNA transcript can be translated into protein, the introns are removed and the exons are spliced. Alternatively spliced exons produce isoforms, which are different representations of the same transcribed gene.

Therefore, the expression of genes deals with the instantiation of different forms of RNA transcripts whose abundances bear on the production of proteins, among other things, which are fundamental molecules that carry out most catalytic functions in cells.

4.3 Profiling transcripts: RNA sequencing

Information about the content of RNA in a biological sample at a given time can be obtained by applying high-throughput sequencing to complementary DNA sequences (cDNA). RNA-seq is the set of experimental procedures that generate cDNA, derived from RNA molecules, upon which high-throughput sequencing is applied. All methods for RNA-seq ultimately pursue the same goal, which is to investigate the transcriptome of cells. The transcriptome refers to the set of all transcripts in a cell, and their abundances, at a specific developmen-

tal stage or physiological condition. RNA sequencing technologies enable the study of the transcriptome to elucidate its complexity and for understanding development and disease [129]. The spectrum of applications leveraging RNA-seq is on the rise [26, 130, 131]. Some key goals of transcriptomics are to catalog the complete repertoire of RNA transcripts; to identify and quantify alternative splicing; to determine the transcriptional structure of genes; and to quantify the changing levels of gene expression under different conditions [129, 132].

4.4 RNA-seq challenges

RNA sequencing is technically more challenging than regular DNA sequencing, and is often a biased procedure. In fact, in RNA-seq protocols practically all steps are potential sources of bias [133]. To sequence the transcriptome the RNA material must be prepared for the task. Different protocols to prepare RNA for sequencing are devised for specific purposes [134], and they depend on the target application. The accuracy of RNA detection depends largely on the nature of the library construction protocol [135]. A library consists of biological material of interest prepared for sequencing. Specifically, a library is a collection of DNA fragments that are ready for high-throughput sequencing with a specific protocol [136].

The construction of RNA-seq libraries involve several steps whose manipulation can complicate the profiling of transcripts. First, RNA molecules are extracted, and the specific type of interest isolated, from the biological sample. The subset of isolated RNA, for example mRNA, is manipulated to make the molecules suitable for sequencing. This is done by fragmenting them into smaller chunks and converting the RNA into complementary DNA through a process called reverse-transcription. The cDNA, or complementary DNA, is the DNA of genes without introns [137].

Fragmentation can be done on RNA or cDNA with different biases in the outcome [129]. Short cDNA fragments are required for sequencing but they later pose computational challenges; reconstructing back sequences from shorter ones is more complicated because they are more ambiguous, in the sense that they can more easily match to multiple locations in the

genome; sequencing errors and polymorphisms in short cDNAs can present also problems for alignment; further, aligning sequences that come from fragments that span spliced junctions is a difficult problem [138].

Following fragmentation and reverse-transcription sequencing adaptors are added to flank the short cDNA fragments. Fragments are amplified to produce thousands to millions of copies of each one and then are randomly sampled. Typical RNA-seq libraries are dominated by transcripts from the most abundant expressed genes, which is the desired outcome for gene expression studies. However, for other type of studies normalization is required to even out the abundance of transcripts [134].

Amplification is a notorious source of bias but its application is generally necessary because of the limited amount of input material [133]. Many short cDNA fragments that are identical to each other can be produced after amplification. This outcome is indistinguishable from the actual abundance of RNA in a biological sample or from an amplification artifact. A way to circumvent this problem is through the use of different biological replicates to determine whether the same cDNA sequences are observed in the library [129]. After amplification the RNA-seq library is ready for sequencing. Then, as it is usual, each cDNA fragment can be sequenced by one end or both ends to produce single-end or paired-end reads.

Transcription activity varies greatly across the genome and the amount of sequencing required for a given sample is not straightforward to compute. It depends on the goal of the RNA experiment and the biological question being asked of the data [129, 135, 139]. Generally, more sequencing depth is required to discover more transcripts in larger genomes, wich have more complex transcriptomes. Experiments that aim at comparing transcriptional profiles may require less depth of sequencing but more replicates [139, 140, 141] than other experiments whose purpose is to discover novel transcripts or to quantify the expression of a particular gene isoform. An ongoing effort to provide guidelines and best practices for RNA-seq is led by the ENCODE consortium [142]. The library preparation workflow for RNA sequencing is schematized in Figure 4.4.

In summary, the RNA-seq protocol outputs reads from the ends of a random sample of fragments in a library [128]. From there, subsequent computational analysis follow to investigate RNA transcription.

4.5 Transcriptome analysis

The application of high-throughput sequencing for RNA discovery presents several computational challenges for transcriptomics. First, it is the problem of gathering together reads into units that we can reasonably assume as transcripts. Then, for gene expression in particular, the next problem is to estimate the expression level of the transcripts found in the previous step. In gene expression there are three primary challenges [132, 143]. The first is to find the location, or likely location, from where a read originates in the genome through the alignment of reads to a reference; a classic problem in bioinformatics and a typical core step in omics pipelines. After reads are mapped to a reference, the second challenge is to piece them together them into transcription units so as to identify the transcripts and isoforms that are expressed. This step reconstructs the set of transcripts present in the sample, and it is referred to as transcriptome reconstruction. Lastly, the expression level of reconstructed transcripts can be estimated by quantifying their relative abundance from mapped fragments. This is called the quantification of relative transcript abundances or expression quantification.

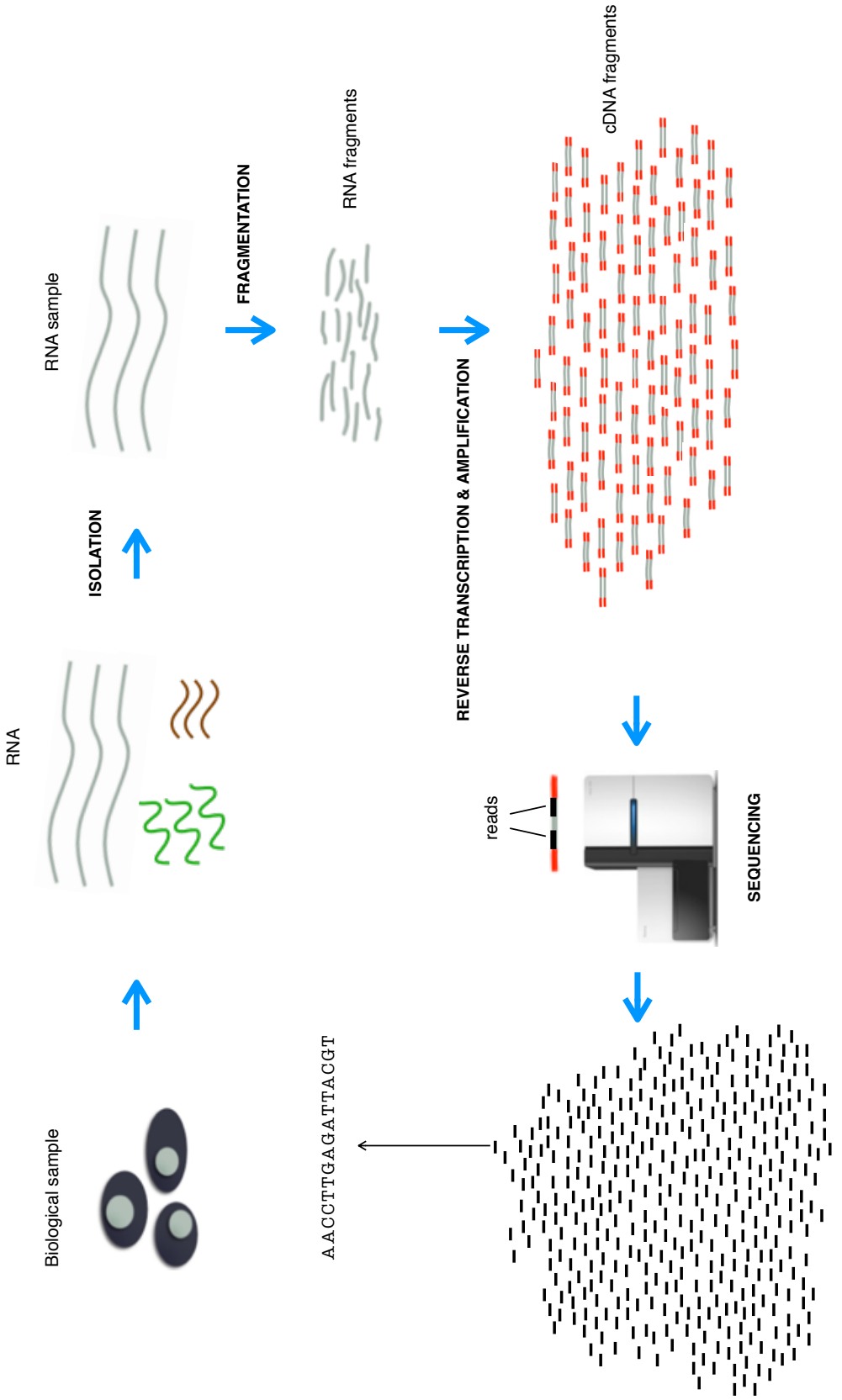


Figure 4.4 – A generic RNA-seq library construction protocol. Steps for preparing biological material for RNA sequencing.

4.5.1 Read alignment

RNA-seq reads are different from conventional DNA sequence reads. Namely, they are generally smaller in length (36-125 nucleotides long [132]) and can reveal splicing if coming from exon-exon junctions. The general challenge is to map millions of short reads accurately and in a reasonable time, while allowing for errors and structural variation [144], and RNA editing. Aligners, not only those used in RNA-seq, allow for approximate matches, and the level of approximation depends on how permissive the tool is with discordances between the read and reference sequence. Some exploration in the parameter space of aligners is recommended to improve their effectiveness. However, for optimal parameter setting and performance expert advice from the developers is usually needed [145].

The main challenges in aligning RNA-seq data come from the types of reads the protocol produces. RNA-seq reads of mRNAs can be of two types: single exon reads or exon-exon-spanning reads [136, 138]. Consequently, many reads can map across splice junctions, spanning exon-exon boundaries; also many different transcripts that represent isoforms from the same gene can be present. Figure 4.5 shows the two types of reads. Read 1 is an exonic read, and maps fully to an exon. Read 2, a junction read spanning an exon-exon boundary, has to be split to be aligned properly.

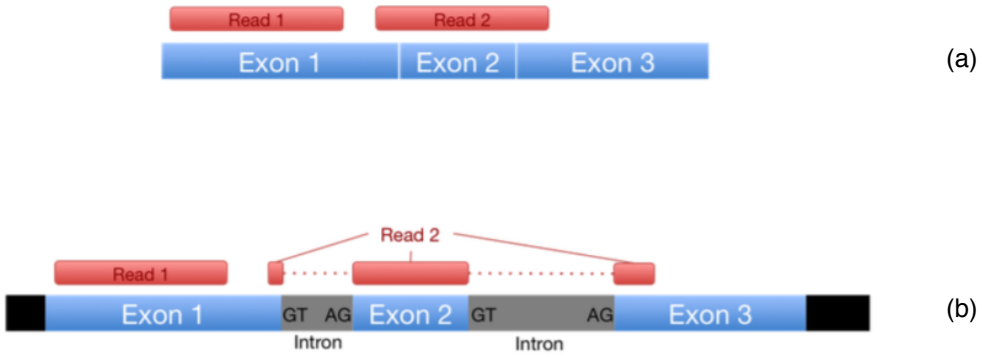


Figure 4.5 – Alignment of short RNA-seq reads to a reference sequence. (a) Read 1 is an exonic read, and Read 2 is a junction read spanning three exons; (b) the challenge is to place spliced reads across introns and correctly identify their boundaries. Image borrowed from [136].

Currently there are two general approaches to identify the likely location in a reference se-

quence where short RNA-seq reads originate. One is by quasi-mapping/pseudoaligning reads to a reference transcriptome; the other is through standard read alignment to a reference genome supplemented with information from an existing gene annotation, if available. As for pseudoalignment, in this approach reads are assigned directly to transcripts from which they likely come from with no exact description on how the reads align to such transcripts [146, 147]. The most popular methods leverage on traditional alignment strategies and make them aware of splicing events in reads, hence called splice-aware aligners. They usually use an annotation of known transcripts, the reference transcriptome, as additional source of information to help in deciding the placement of spliced reads, and to identify possible novel splice junction sites.

Alignment performance varies with the complexity of the genome under study, and it is impacted directly by the aligner's parameters settings. Popular tools have been found to underperform for most metrics because they are commonly set to default settings [145]. In general, a best overall approach to alignment cannot be called since performance is evaluated for specific data, with respect to specific measures and for a target application. As a result, benchmarking studies are proposing combinations of high-accuracy tools in a sort of "cocktail" to determine appropriate pipelines for the analysis of RNA-seq data [130].

4.5.2 Transcriptome reconstruction

Gene expression is the estimation of transcript abundances that are present in a sample. To quantify abundances, full-length transcript sequences need to be put together first by piecing short RNA-seq reads. The intrinsic transcriptome complexity, as manifested by alternative splicing, polymorphic events and dynamic expression levels, along with limitations in sequencing technologies make resolving the structure of transcript sequences a major challenge [135].

The problem of reconstructing transcripts is fundamentally dealt with in two ways, with strategies that are guided by a reference sequence, and by those who are not. Reference-based

approaches assume the availability of a reference sequence that serves as the structure upon which the target transcriptome is built. Short RNA-reads are aligned to a reference genome using a splice-aware aligner, and grouped into gene regions. Reads aligned to each gene locus are then parsimoniously assembled to discover as many isoforms as needed to explain the data [148, 149].

It is clear that the quality of the reference sequence impacts the reconstruction of transcripts, and misassemblies in the reference may lead to faulty or incomplete reconstruction of transcript sequences. The reference also acts as a template to remediate small gaps within transcripts when there is lack of read coverage. In addition, this type of reconstruction allows, in principle, the discovery of novel transcripts (that is, isoforms of genes or splice junctions) [150], whose levels of expression are generally low and thus not reported in the reference transcriptome annotation. However, lowly expressed isoforms may be supported by few reads in their specific splice junctions, and few reads in splice junctions are more likely to be considered as false positives. This results in a bias toward the discovery of novel transcripts who are strongly expressed [136].

In addition, reference-based approaches can miss transcripts structure following missalignment of splice reads. For example, a splice read coming from a transcript spanning large introns may be unaligned, or misaligned due to constraints in the alignment search to accommodate the sequence. Also, ambiguities in the placement of reads that align equally to multiple genomic locations can produce transcripts that generate from regions that do not correspond to transcription sites [150].

The second way to transcript reconstruction is 'unguided' and does not require a reference sequence; assemblies of this sort are called 'de novo'. These approaches rely and exploit the redundancy of short RNA-seq reads, find overlaps between their sequences, and reduce them to unique transcript sequences. The core challenge for this type of reconstruction is to partition the reads into disjoint components, which determines the splice sites, to represent all isoforms of a gene [132]. Major challenges in de novo transcript assembly are the discrimination of

sequencing errors from natural variations, the tradeoff between the complexity of the overlap graph and the sensitivity of the assembly, and to distinguish highly similar transcripts, like those originated from different alleles [132, 150].

An hybrid approach combining strategies to reconstruct transcript sequences either by first aligning and then assembling, or the opposite, by fist assembling and then aligning, can produce a more comprehensive view of the transcriptome structure. Reconstructing transcripts in this way captures known information as well as novel variation [132]. In the presence of a reference sequence, transcripts are first reconstructed from aligned reads and then assembled; assembly can be guided by transcripts obtained from the alignment and/or applied to reads that failed to align. Conversely, if a high-quality reference sequence is not at hand or only related sequences with enough similarity are available, assembly is commonly preferred prior to alignment; then alignment of assembled transcripts and unassembled reads follow.

The choice of reconstruction strategy depends primarily on the data available and the purpose of the study. Nevertheless, finding the structure of transcripts and reconstructing gene isoforms from spliced reads remain the most prevalent problem of RNA-seq data [136].

4.5.3 Expression quantification

The process by which functional products are generated from genes is referred to as the "expression" of a gene [143]. Quantifying the expression of genes in a sample, given short RNA-seq reads, means estimating the relative abundances of reconstructed transcripts, which correspond to the counts of reads aligned to them. That is, reads aligned to features (gene, exon, isoform of a gene) in an RNA-seq experiment are used as a proxy to measure the expression of that feature. A pool of sequence reads is sampled uniformly such that the expression of features is represented proportionally [149]. The number of reads aligned to a feature depends on the feature's expression, the feature's length, the sequencing depth and the expression of other features in the sample [136]. RNA-seq does not allow for absolute measurements of expression levels because read counts cannot be compared directly between features in

the same biological condition or across different conditions; they can only be compared proportionally. Thus RNA-seq is a relative abundance measurement technology. For example, two features A and B in the same condition, where B doubles the number of aligned reads in A. Feature B can be expressed twice as much as A; or feature B can be twice as long as A, and expressed with the same number of reads as A; or perhaps features A and B are expressed equally but there is another feature whose sequence is close to A's and unique read alignments to A are not possible. To compare expression levels of features within the sample, read counts are adjusted by normalizing for each features' length and for the reads sequenced. This scaling normalizes read counts in units representing the proportion of transcripts in a pool of RNA-seq mapped reads [143, 151, 152, 153].

There is less complexity in comparing the expression of the same features across biological conditions than different features in the same condition. This is in the sense that it suffices to compare differentially the expression of the same feature between the taget conditions. The expression of a feature, a gene, between two conditions, will be measured with different read counts, and their differences will reflect real biological differences or differences because of protocol noise.

To compare the expression of features between two conditions, read counts are normalized with the goal to remove systematic effects that are not associated with the biological differences of interest [136]. Normalization consists in calculating the number of reads for each feature relative to the library size (number of aligned reads obtained from sequencing the library, that is, the sequencing depth), and with respect to the total RNA repertoire expressed in the biological sample. The certainty in the true expression of a feature increases as more data are available. In the same way, variations due to contamination can be more robustly detected with more data. Therefore, multiple measurements, using biological replicates for each condition, are to be made to identify the expression levels and associated variations of the features across conditions. A biological replicate is an RNA sample from an independent growth of cells/tissue [142], which shows the biological variability of the system under study.

The number of replicates depends on the purpose of the study. The recommendation is to use at least six replicates per condition to identify differentially expressed genes, and at least twelve to identify as many expressed genes as possible [154].

In addition to normalization, a common representation for comparative analysis of read counts between expressed features in a multi condition setting is the logarithmic fold change. More precisely, normalized read counts ratios are transformed to the logarithmic scale, usually logarithm base two. This has the advantage of interpreting the expression of features in terms of doubling values and also in treating changes in expression symmetrically. For example, a gene upregulated by a ratio or factor of 2 has a log2 fold change of 1 ($\log_2 2 = 1$); a gene downregulated by a factor of 2 has a log2 fold change of -1 ($\log_2 \frac{1}{2} = -1$); and a gene expressed at a constant level, that is, without expression change between conditions, has a log2 fold change of zero ($\log_2 1 = 0$) [155].

4.6 Differential gene expression

Given RNA-seq reads from two different conditions and reconstructed transcript sequences, the goal of differential gene expression is to predict which transcripts have different abundances between said conditions. Based on read counts from replicated biological samples, two are the tasks performed by all differential expression tools [136]: calculate the fold change of read counts to represent the magnitude of differential expression, and estimate the significance of the difference.

To estimate statistically significant genes, tools for differential gene expression make assumptions about the form of the underlying read count distribution, and on the capacity to accurately measure the mean and variance of read counts for each gene [154]. Fundamentally, however, the computation of differential expression rely on the assumption that the expression levels of the transcriptome across conditions remain mostly unchanged, that is, that most genes between conditions are not differentially expressed. If this assumption is not met by data, both indicators of relevant expression (log2 fold change and significance measure) are

likely incorrect [136, 154, 156].

4.7 Differential gene expression and lossy compression of quality scores

High-throughput sequencing of RNA molecules has enabled the quantitative analysis of gene expression at the expense of storage space and processing power. To alleviate these problems, lossy compression methods of the quality scores associated to RNA sequencing data have recently been proposed, and the evaluation of their impact on downstream analyses is gaining attention. The following sections present a first assessment of the impact of lossily compressed quality scores in RNA sequencing data on the performance of some of the most recent tools used for differential gene expression. This work was developed in collaboration with Leibniz University Hannover, and is published in [157].

4.7.1 General context

High-throughput RNA sequencing (RNA-seq) is undergoing rapid evolution since its introduction back in 2008 when several research groups, encouraged by the accessibility of novel high-throughput sequencing technologies, set out to study the transcriptome of different organisms [153, 158, 159, 160]. It is through nucleotide sequences of RNA that information encoded in an organism's DNA is made available to the cell, and that it can be interpreted by the cell to guide the synthesis and regulation of proteins. The RNA sequences are gene readouts, i.e. copies of gene regions of DNA. These gene readouts are called transcripts and the set of all the transcripts present in a cell, or a population of cells, at a given time constitutes the transcriptome.

Researchers can gain a better understanding of the workings of cells and their connection to diseases by investigating the levels of gene activity in the transcriptome. The activity of a gene is the result of a process known as gene expression through which the DNA nucleotide sequence of a gene is converted into nucleotide sequences of RNA, and then into the amino

acid sequence of a protein; though it is not always the case that RNA sequences lead to protein sequences. The amount of gene activity can be measured by estimating the number of transcripts in a tissue sample. RNA-seq data is widely used to get quantitative information on the differences in the expression of genes between a test and control conditions. However, gene expression levels are very fragile and reflect uncertainties associated with sampling as well as technical and biological variance [161]. The certainty about the observation of a gene expression level can be improved by increasing the number of sequenced reads in a condition, which can be achieved by adding biological replicates and by deeper sequencing of existing replicates [140].

The test for differential gene expression (DGE) relies on the estimation of transcripts across conditions, which requires the reconstruction and quantification of millions of sequenced reads. The high computational cost associated to the storage and processing of millions of reads is shared by all functional genomic assays driven by high-throughput sequencing. The wealth of raw sequenced data, and the complexity of measurements to be inferred make the setup of a working bioinformatic pipeline a challenge, and an assessment of its accuracy is difficult [110, 109, 111]. Moreover, the situation aggravates in applications like DGE where multiple, deeply sequenced samples need to be analyzed. In recent years several research groups have investigated methods to improve the effectiveness of compression technologies for the storage of high-throughput sequencing data. In particular, approaches to lossy or quasi-lossless compression of quality scores have received special attention [45, 85, 88, 87], along with an interest to measure their impact in the calling of genomic variants [42, 114], so far the sole downstream application tested for evaluation.

In the context of gene expression, this work sets out to explore the effect of lossy compression of quality scores. For this purpose we start by observing its effect on transcript reconstruction over a simulated sample with different depths of coverage. Then, we take two real datasets of RNA-seq data and run them on a state-of-the-art DGE pipeline, and provide a first assessment of the impact. In particular, the goal is to understand if differences arise in the calling of expressed genes, between a two-condition DGE pipeline that features full quality score scale

of RNA-seq data, and a pipeline featuring reduced resolution. The focus is only on significant genes with the strongest activity and state-of-the-art tools are used to build the pipeline.

In summary, this work shows:

- That lossy quality scores marginally affect the reconstruction of transcripts in simulated data, a result that is corroborated in the calling of genes in the test for differential gene expression
- The application of lossy compression in a pipeline, in which transcript reconstruction use quality scores, testing differential gene expression
- How high rates of lossy compression of quality scores in RNA-seq data do not compromise, in principle, the calling of significant genes when testing for differential gene expression in a two-condition setting.

4.7.2 RNA-seq and differential gene expression

RNA-seq functional assays have the primary goal of quantifying abundances of mature molecules of messenger RNA (mRNA) in a cell. Different types of RNA molecules are produced during transcription but only mature mRNAs will be translated into proteins. In eukaryotic cells, splicing happens cotranscriptionally in mRNAs molecules: a process where all intron sequences are removed from mRNA transcripts and the remaining exons are joined to form a continuous sequence. Splicing can occur in different ways leaving in or out exons from the final transcript. The possibility of different splicing patterns from the same mRNA transcript is called alternative splicing and it allows the production of different proteins from the same gene during translation.

Broadly speaking, RNA-seq applications can be grouped in two categories. When the expressed transcripts are used to conduct transcriptome annotations, the application is qualitative. Other applications require some form of measuring and thus they are considered as quantitative. Examples of these applications are: the quantification of novel transcripts, alternative splicing

and gene expression. The goal of most RNA-seq experiments is to identify genes whose expression change across two experimental conditions. These differential gene expression experiments require at least six biological replicates per condition with sufficient sequencing depth [140, 154, 162].

The RNA-seq protocol is somewhat the same across platforms: samples of RNA are isolated, copied into complementary DNA, amplified and sequenced to obtain reads. The workflow described below reconstructs the transcriptome from the resulting reads and measures the expression of genes by quantifying read abundances.

From here, and until the end of the chapter, I will borrow the term "assembly" to refer to the idea of piecing together aligned reads into full and partial transcripts. This nomenclature is sometimes used in protocols for transcript-level expression, as seen in [148].

Figure 4.6 shows how a pipeline for DGE can be structured in three steps:

- Assembly. Spliced aligners like TopHat2 [163] and HISAT2 [164, 148] can map exonic reads and identify splice junctions from reads spanning different exons. However, the assembly of exon-spanning reads requires an additional tool. According to [165] the best performing tools for this task are Cufflinks [166] and StringTie [167]. The reconstruction of the transcriptome is complete when both exonic reads and exon-spanning reads are mapped.

- Quantification. Aligned reads are counted to measure the expressed genes in the reconstructed transcriptome. Cufflinks and StringTie simultaneously assemble and count the reads mapped to each transcript. Lightweight approaches such as Sailfish and its successor Salmon [147] and Kallisto [146] bypass the assembly step and directly estimate the read count by pseudo-aligning to the reference transcriptome.

- Estimation of magnitude and significance of differential expression. The count of reads is a relative value of the sample. Its value depends heavily on the amount of fragments sequenced and the effective length of the genomic region in an RNA-seq experiment.

Therefore read counts should be normalized to compare features, like genes, within a sample. For absolute expression, common units for normalized read counts are transcripts per million (TPM) and fragments per kilobase of exon per million reads mapped (FPKM).

The magnitude of differential expression between two or more conditions is estimated by computing the fold change of normalized read counts from replicated samples. DGE tools make assumptions about the distribution of the read counts to determine the genes whose expression varies between conditions. These tools estimate the significance of expression differences by testing the null hypothesis that a gene's expression between conditions (e.g. treatment vs. control) is unaffected. Several publications [154, 156, 168] have reported overall best performing tools for estimating DGE and some consensus exist on the tool DESeq2 [169].

4.7.3 Experimental setting

In our first setting we investigated the effect of lossy compression of quality scores on transcript reconstruction. Using the Flux simulator [170], we generated three samples of the human chromosome 22 with one, five and ten million reads and ran them through HISAT2 and StringTie to assemble the transcripts. The samples were input in four modes: with and without quality scores, and after applying lossy compression with the tools Quartz [87] and P-/R-Block [88]. We evaluated the reconstruction of transcripts by means of the average per-base-coverage. Because we used simulated data, the reference coverage is known and after assembly the coverage for reconstructed transcripts can be computed.

In our second setting, we focus on determining differentially expressed genes on replicates of RNA-seq data. The layout consists of three steps: assembly, quantification and the test for differential expression (see Figure 4.7). The sequenced reads are first mapped to the reference transcriptome guided by the genome annotation during assembly. The mapped reads are then analyzed to reconstruct the possible transcripts from which they came from; the computation

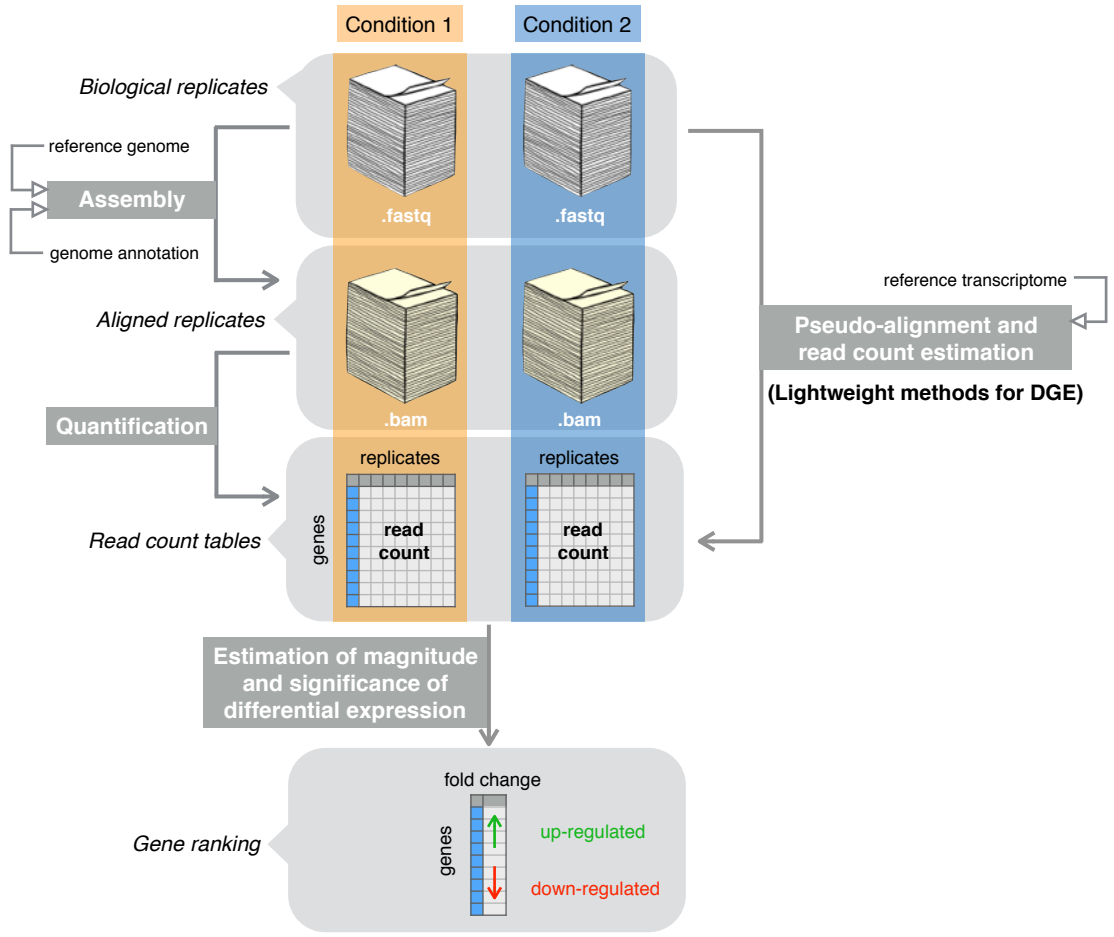


Figure 4.6 – Organization of a pipeline for differential gene expression.

of abundances of reconstructed transcripts follows. Both the assembly and quantification steps are repeated for each replicate in every condition. The test for differential expression takes place after the abundance count of all replicates of all conditions has been obtained. In this last step the magnitude and significance of expressed genes are estimated.

In the pipeline of Figure 4.7 a pre-processing step is added where lossy compression is applied to the quality scores of an input replicate (see Figure 4.8). In this step the sequences of nucleotides are kept intact, but their quality scores are compressed with controlled loss of information, and ultimately transformed to a coarser resolution after decompression. Three methods of lossy compression of quality scores were applied: a uniform quantization with 2 and 8 bins (UQ2, UQ8), and the approaches proposed in the Quartz [87] and P-/R-Block [88].

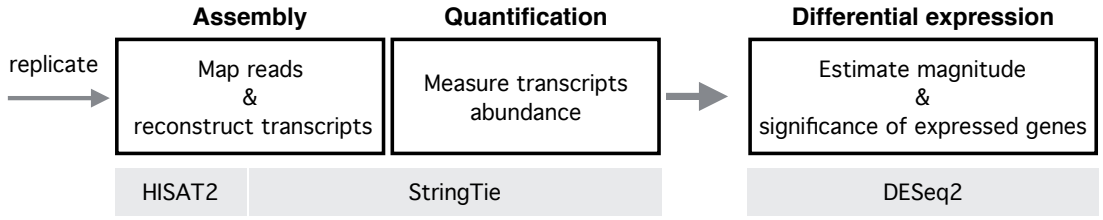


Figure 4.7 – Steps for determining differentially expressed genes on replicates of RNA- seq data. The assembly and quantification steps are repeated for each replicate in every condition. The name of the tools used are stated below each step.

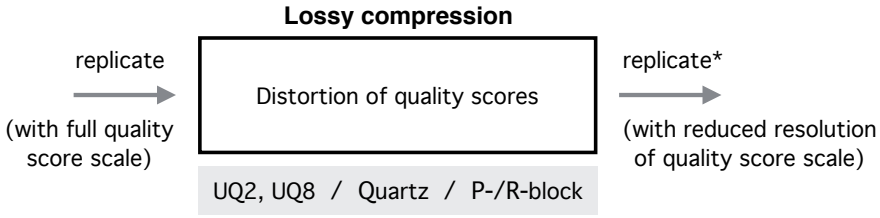


Figure 4.8 – Lossy compression of quality scores for each replicate is prepended to the DGE pipeline. Three methods are used: uniform quantization, Quartz and P-/R-Block.

The pipeline under test is summarized in Figure 4.9. Tests on this pipeline were conducted for two organisms: the yeast S. cerevisiae [171] and the MCF-7 human breast cancer cells [172]. For each, a total of twelve replicates (six replicates per condition) were used. The results are presented in the following section.

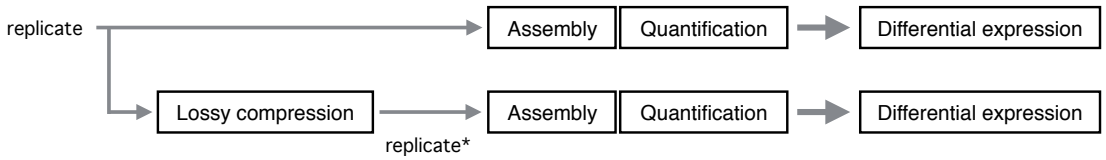


Figure 4.9 – Layout of the pipeline for differential expression with lossy compression of quality scores. This pipeline was run for three lossy compression methods on RNA-seq data for two organisms.

4.8 Results

In Table 4.1 we report the overall alignment percentage for four modes of three simulated samples of the human chromosome 22. Along with the alignment rate, the bits required per quality score is shown; the theoretical lower bound for this rate (0 bits/QS) is shown.

Table 4.1 – Overall alignment rate percentage with HISAT2. This value is the sum of the percentage of reads aligned exactly one time plus the percentage of reads aligned more than one time.

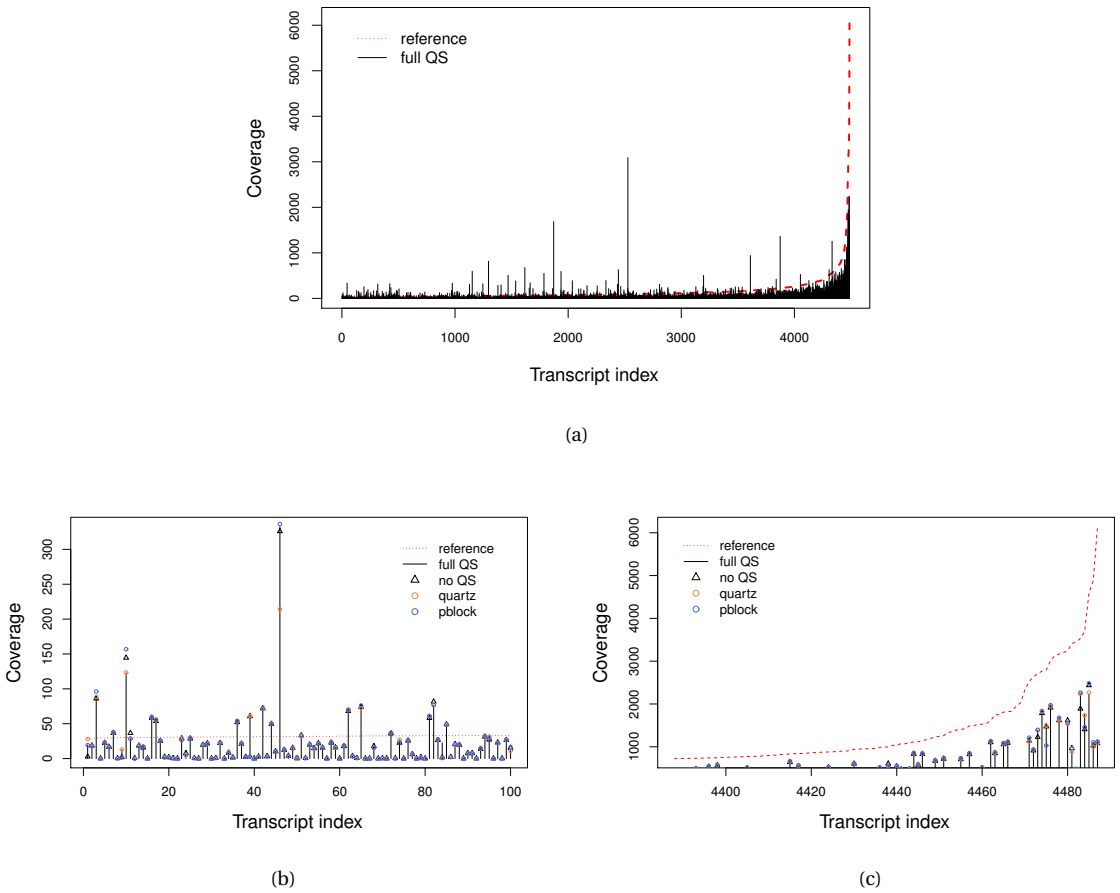
		1M	5M	10M	bits/QS
	full QS	77.77	78.28	79.63	3.16
	no QS	76.5	77	78.25	0
Lossy compression	quartz	77.37	77.56	79.29	1.12
	pblock	78.73	78.91	80.61	0.98

The distribution of transcripts ordered by coverage is shown in Figure 4.10(a). This data reports the coverage per reconstructed transcript in the file with 10 million reads and with full quality score scale. Figure 4.10(b) and (c) show in detail the coverage for the bottom and top 100 transcripts. We observe how the fluctuation of coverage is marginally different between the four modes under test.

In the analysis of gene expression the measure of change is usually reported in terms of the fold change estimate. This value represents how much the expression of a gene seems to have changed between conditions. The fold change can be positive or negative and it is commonly transformed to log2 scale; for example, a gene with a log2 fold change of 1 means that the gene's expression increased by a factor of $2^1 = 2$. Positive values of fold change signal upregulated genes and negative values signal downregulated genes.

To determine the significance in the calling of expressed genes the method for differential analysis of count data proposed in DESeq2 [169] was used. For every gene a hypothesis test is conducted to decide against the null hypothesis that the variability observed of a gene's expression between conditions is the same; the result of the test is reported as a p-value. These p-values are corrected for multiple testing and adjusted to account for false positives. The false discovery rate statistic can then be used to set a threshold on the allowed percent of false positives in the set.

For the performed tests a false discovery rate of 10% was considered and the result was sorted by the log2 fold change estimate to obtain significant genes with the strongest up- and down-regulation. The last step of the pipeline shown in Figure 4.9 outputs a list of ranked genes.



(a)

(b)

(c)

Figure 4.10 – (a) Coverage of chromosome 22 in the file with 10 million reads. (b) Bottom and (c) top 100 transcripts in the same file.

The goal of this work consists in measuring if the calling of significant genes with the strongest up- and down- regulation in a DGE pipeline is affected by lossily compressing the quality scores associated to RNA-seq data. To get a first assessment of the impact the ranked lists computed by the pipeline for every lossy compression method were compared. In table 4.2 the median compression rate in bits per quality score is shown for every compression method along with its confidence interval.

Tables 4.3 and 4.4 show the ranked lists of log2 fold changes and the associated genes; values in bold are log2 fold changes for whose gene calling was different from the genes indicated in the rightmost column.

Table 4.2 – Median compression rates in bits per quality score. The values are reported for both organisms and for each condition and lossy compression method.

	cond		UQ2	UQ8	Quartz	P-/R-Block
yeast	1	3.075	0.2	0.735	1.75	1.015
		[3.05, 3.10]	[0.2, 0.21]	[0.72, 0.75]	[1.66, 1.89]	[1.0, 1.04]
	2	3.08	0.205	0.735	1.025	1.015
		[3.05, 3.09]	[0.2, 0.21]	[0.72, 0.75]	[1.75, 1.85]	[1.0, 1.03]
MCF-7	1	2.21	0.16	0.70	0.57	0.975
		[1.49, 2.47]	[0.07, 0.19]	[0.35, 0.82]	[0.46, 0.61]	[0.52, 1.13]
	2	1.68	0.09	0.44	0.49	0.635
		[1.59, 1.95]	[0.08, 0.12]	[0.4, 0.57]	[0.48, 0.55]	[0.58, 0.80]

Table 4.3 – Ranked list of log2 fold changes for the yeast and genes associated.

	regulation	log2 fold change					gene
			UQ2	UQ8	Quartz	P-/R-Block	
yeast	up	6.0629	6.0574	5.9761	6.0631	5.9764	YOR192C-A
		5.7313	5.8074	5.8105	5.8147	5.8108	YDR034C-C
		3.6137	3.5778	5.0871	5.2070	3.5193	YHR214C-C
		2.8025	2.7971	2.7996	2.8031	2.7980	YPL025C
		2.5757	2.5702	**2.6641**	2.5764	2.5716	YOR376W
		2.4249	2.3629	**2.5722**	2.3671	2.4629	YPR158C-C
	down	-8.0886	-8.0846	-8.0834	-8.0899	-8.0844	YOR192C-B
		-8.0082	-8.0026	-8.0032	-8.0103	-8.0080	YDR034C-D
		-6.2723	-6.3004	-6.1566	**-6.6860**	**-6.1452**	YER160C
		-3.4012	-2.8554	**-6.0406**	**-6.4943**	**-6.1324**	YHR214C-B
		-2.4985	-2.5184	-4.5319	**-4.8144**	**-3.0414**	YDR210W-A
		-1.8940	-1.8929	-2.4752	**-2.5104**	**-2.5042**	YKL078W

The results are discussed in the following section.

4.9 Discussion

We set out to explore the effect of lossy compression of the quality scores associated to RNA-seq data in a pipeline for differential gene expression. The overall effect reflects on the calling of differentially expressed genes, which is ultimately the output of interest for this omics

Table 4.4 – Ranked list of log2 fold changes for the MCF-7 and genes associated.

	regulation	log2 fold change					gene
			UQ2	UQ8	Quartz	P-/R-Block	
MCF-7	up	5.2348	5.2421	5.2368	5.2324	5.2430	NM_144967
		4.2312	4.2329	4.2319	4.2312	4.2329	NM_014668
		3.8070	3.8309	3.8114	3.8058	3.8430	NM_001555
		3.7533	3.7575	3.7543	3.7516	3.7580	NM_002614
		3.6763	**3.6962**	3.6822	3.6759	3.6863	NM_001170961
		3.5690	**3.6856**	3.6276	3.5676	3.6715	NM_001202474
	down	-7.4730	-7.4970	-7.4778	-7.4722	-7.5012	NM_138780
		-4.9594	-4.9775	-4.9588	-4.9590	-4.9777	NM_001102594
		-4.2973	-4.3204	-4.3020	-4.2963	-4.3232	NM_001207059
		-3.5473	-3.5865	-3.5552	-3.5459	-3.5901	NM_014309
		-3.4331	-3.4554	-3.4369	-3.4323	-3.4581	NM_017851
		-2.5689	-2.5736	-2.5697	-2.5630	-2.5743	NR_131192

application.

The calling of significantly expressed genes can produce a large number of hits, particularly in comprehensive studies with complex biological data [173]. It has been reported, however, that from the pool of differentially expressed genes only a portion are highly expressed [174]. We focused only on higly expressed genes in order to simplify our assesment but also to report concretely the impact produced by applying a lossy representation to the quality scores. As such, the strongest up- and down- regulated genes served to summarize the effect of lossy compression.

We observed small changes in gene regulation, as per the log2 fold value reported in Tables 4.3 and 4.4, after the application of lossy representation in both datasets. Given that it is only during transcript reconstruction that quality scores are used, the changes originate at this point in the pipeline.

In the human data these changes were only present when applying the most severe compression method, while several log2 fold values changed for almost all compression methods for the yeast data. We note that these changes (values in bold in Tables 4.3 and 4.4) were sufficient to impact the ranking order of the expressed genes. However, the silver lining is that while the ranking value changed, as measured by the log2 fold value, the set of highly expressed genes remained the same. Thus, in our tests, the calling of highly expressed genes whether up- or down- regulated was preserved following the aplication of a coarser representation for the quality scores.

It is clear that the strategy for lossy representation impacts differently the measurement of log2 fold value. Furthermore, it is not possible to recommend one strategy over another or to suggest that one is better than another. For example, the results for downregulated genes in human data seem to suggest that all compression methods perform transparently, as no changes in regulation were noted. However, for downregulated genes in the yeast sample the compression strategy seems to bear on the result and straightforward strategies to lossy representation (uniform quantization) seem to fare better.

At this level of analysis we can say that the calling of the most expressed genes are slightly affected by lossy quality score representation, a result that seems to suggest that finer inspection in processing the quality scores is required.

The task of identifying expressed genes relies ultimately on reconstructing transcripts from aligned reads, the entry point to abundance quantification and differential expression of genes. Figure 4.10 shows the effect of lossy compression in terms of reads aligned to each identified transcript in a simulated sample; the expected coverage, as reported by the simulator, is marked with red dashes. Overall, we observed marginal changes in read coverage, a direct consequence of changes in alignment percentages after quality score compression. This has in turn an effect on the quantification of abundances.

The changes in alignment percentages resulting from lossily representing the quality scores provided a clear hint to examining the impact of lossy representation on alignment. This is the subject we will explore in the next chapter.

5 Lossy quality scores and reference-based alignment

In the last two chapters we have surveyed two omics applications that have accrued a lot of attention due to their relevancy and scope of applicability. Although the technology upon which they rely is in active improvement, efforts in keeping up with and aid to its betterment have produced opportunities for intense bioinformatic tool development. Along with this, a sensible and pertinent calling to the systematization of methods and procedures to conduct experimentation and analysis is increasingly becoming a pressing issue [175, 36, 47]. Efforts in the right direction are the publication and active actualization of data processing guidelines and best practices like those initiated by GATK and ENCODE groups [176, 177].

5.1 Challenges in omics applications with lossy quality scores

With the increasing number of omics applications leveraging on sequencing technology, and the specific nature of the computational methods devised for them, along with the overwhelming collection of tools developed for them, adopting a particular pipeline for analysis may be argued limited and idiosyncratic. To cope therefore with the deluge of tools and methods for omics applications, we are witnessing solutions in the type of overarching frameworks [130] that support the latest best practices for widely used “seq” analyses [178]. In addition, the ongoing trend is toward the standardization of pipeline descriptions via dedicated workflow

languages such that pipeline components, their connections, and configuration parameters, can be specified and modified accordingly [179].

Still, the choice and setup of a suitable pipeline is contingent upon the application of interest. Previously we explored pipelines for calling genetic variants, and differentially expressed genes, as our investigation pointed to the rapid adoption of these applications following the advent and progress of whole genome and transcriptome sequencing using next-generation technologies [26, 60, 70]. As the goal is to investigate the relevance of quality scores in preponderant downstream applications, we consider that this choice was appropriate.

Lossy quality score representation was originally investigated in the calling of genetic variants [37] using an all-comprising software package [180] organized much like GATK's current pipeline for variant calling. It was suggested that given the inherent errors in calling SNPs, the calls were inherently robust to errors. The impact of lossy quality scores was measured minutely by examining the positions of discrepant SNPs, and their concordance with the public database for single nucleotide variations[1]; the corresponding alignment coverage was also accounted for. It was found that most discrepant calls were product of minuscule variations in the values of quality scores (of one or two units) in regions with low alignment coverage. Such small changes marginally went over the passing threshold to be called a variant, and were reasonably called as such. The continuation of this line of research has been primary focused on exploring alternative representations for lossy quality scores under a similar scheme of evaluation. That is, assessing the effect of lossy compression in a full-fledged downstream application, notably the calling of genomic variants. The evaluation approach is similar to that presented in [41, 86, 88], where the authors reasonably suggest the adoption of commonly used performance metrics (sensitivity, specificity, etc) in day-to-day variant calling. The research in the field of lossy quality score compression has followed suit, as presented in section 3.10.2 and as can be seen in [45, 46, 57], with efforts to systematize the evaluation of the impact [42].

Indeed, the above performance metrics are widely used in variant calling but they do not

[1] https://www.ncbi.nlm.nih.gov/snp/

usually stand alone for validating results. The metrics are commonly complemented with supporting material to explore potential factors that could contribute to discordant results. For example, recent works incorporate in the analysis multiple input datasets by various sequencing platforms, as well as exome capture systems and exome coverage [111]; others support the inclusion of additional metrics that measure factors influencing call concordance, such as local GC content, depth of coverage, mapping quality, repetitive DNA elements, etc [59]; moreover, the increasingly popular integration of multi-omics datasets and techniques for more holistic understanding of downstream analyses is inevitably taking place [181]. What is more, and in the midst of abundant choice of computational tools and pipelines, alternative approaches have opted to source information from multiple variant callers to improve the accuracy of calls with good results [182, 183]. All in all, an additional important source for insight is the manual review [111].

5.2 Fundamental challenges in bioinformatics

In the light of increasingly complex workflows and the aggregation of multiple sources of information to perform downstream analyses, along with the lack of published guidance, there is high variability in the establishment, configuration and validation of bioinformatic pipelines. A recent paper investigated real world experiences across the bioinformatic community, and reported as part of the key insights, the sore lack of standards in bioinformatic workflows as well as in software tools and data management [103]. In the same vein, and to understand existing practices with respect to bioinformatic pipelines, a systematic review of the literature was carried out in [184]. Inconsistencies in methods and validation strategies in pipelines was confirmed across the published literature. To address these problems, an initiative led by the Association of Molecular Pathology outlining the consensus of recommended guidelines for next-generation sequencing bioinformatic pipelines was published recently [184]. This work aligns with recent efforts to harmonize analyses in bioinformatic workflows [47].

The constant evolution of technology enables continual upgrades in computational tools and

methods, making it possible to close gaps to actively improve bioinformatic analyses. It is clear, however, that the pace at which this development is taking place makes it unfeasible to explore all available options. And the lack of consistent guidelines can readily obfuscate analysis results.

It is already the case that the challenge to navigate, and sift through omics applications, computational tools and pipelines, and the configuration/optimization of their parameters, turns rapidly into an overwhelming feat. Moreover, in waiting for incipient standardization initiatives to gain momentum, the identification of proper performance metrics, and assessment of tools and methods systematically to better leverage technology innovation, is a taxing and complicated endeavor [48]. What is more, when surveying tools and methods one should be cautious of the self-assessment trap [185], and select under different evaluation criteria the most appropriate choice for the particular goal in mind. In the evermore complex workflows, the number of processing steps for "deep analysis" [186] is increasing. Projections to the year 2025 estimate that over 75% of the cost and complexity of genomic workflows will be taken over by data analysis and storage [28].

5.3 Analysis of lossy quality scores

As per the application of lossy quality scores concerns, it is arguable that we can obtain clear insights of their effect on the ever-changing omics pipelines without the adoption of clear practices and guidelines. This is rapidly changing though, and in this context, the promotion of principled procedures is already taking place. Earlier this year, a comprehensive review of benchmarking studies of computational omics tools presented a summarization of practices, putting forward principles for rigorous, reproducible and transparent benchmarking [48].

All things considered, we have learned that evaluating the effect of lossy quality scores on relevant omics applications primarily involve:

- The challenging task of sifting through, selecting and configuring appropriate software

tools, organized in a pipeline suitable for the application in question

- The utilization of relevant metrics and principles to guide systematic analysis and comparison

- Identification of the actual steps in the workflow where quality scores are relevant. That is, in order for the analysis to make sense, knowledge of the methods whose output rely on the use of quality scores is required

In addition, while the ultimate goal is to assess the impact of lossy representation of quality scores in a full pipeline, measuring the impact on multi-step pipelines with summarized metrics may be ineffective for the purpose of understanding the role of lossy quality scores. This is because data is transformed continually within the pipeline, and errors and associated uncertainties of the computational methods are combined and shepherded throughout steps and along with the data. As we have observed small magnitudes of variation in assessing the impact, it begs the question if narrower focus on a relevant step in the pipeline would provide more insight to help elucidate and pinpoint the effect of lossy quality score representation.

In the light of the above discussion, we shift gears and take the sensible approach to focus on read alignment to explore the effect of lossy quality scores. Alignment is a fundamental upstream processing step in most next generation sequencing workflows [187], and along with sequence assembly, the problems they tackle have been studied for almost thirty years. In fact, methods for read alignment and genome assembly represent key developments for sequence analysis, so much so that spearheading methods are reported as computation milestones in the history of DNA sequencing technology [70].

5.4 Reference-based alignment

Alignment is a crucial task because it guides subsequent processing steps in the pipeline. In sequence alignment, the nucleotides of two or more sequences are compared to find some degree of similarity between them. The search looks for patterns of nucleotides that are in the

same order in the sequences being compared, and this procedure is applied to every query sequence that is to be mapped, or aligned, to the target sequence. The goal of an aligner is to determine the likely location of origin for each query sequence from a large collection of reference data (also reference sequence, usually a reference genome). The alignment has to be approximate to allow for sequencing errors and true genetic variations. Three elements are needed to carry out sequence alignment: a collection of sequence reads, a reference sequence, and a set of constraints and a distance threshold [188]. The read's likely point of origin with respect to a reference sequence is a read match. A match is called when a substring in the reference meets the imposed constraints and falls within the distance of the query read.

Generally, alignment is carried out in two steps. First, there is a search to find the set of candidate locations between the query read and the target reference. Second, the read is mapped to candidate positions to determine the best alignment locations, complying with specific rules imposed by the aligner. Most alignment methods build auxiliary data structures (indices) for the reference sequence, and sometimes also for the reads, to create rapidly the set of candidate locations in the first step. The idea is to scan sequences against indices to generate seeds, exact matches of part of the read with part of the reference, and compute an alignment score per match. The alignment policy determines what is factored in in the alignment score, which usually allows for read errors, nucleotide deletions and insertions (indels), SNPs and gaps (long indels). The number of available alignment methods has increased in the last twenty years, and nearly all of them have been developed in the last decade for both DNA and RNA-seq data [145, 188]. However, only a portion has found their way toward regular use, presumably because methods are only as good as they are useful, and poor portability or faulty design interface count against their adoption [189].

Aligners are devised to handle large amounts of sequence data, a typical need in high-throughput sequencing experiments. They need to adapt to library protocols and to exploit their features, like utilizing read pairing information in the light of paired-end reads. In addition, aligners need also to leverage advances in sequencing technology for different platforms, and exploit, for example, the length of sequence reads, the error rates for indels and

substitutions, base quality scores, etc [187]. All these reasons motivate the development of new alignment methods, and with the growing number of biological applications, we can expect their further specialization. The selection of an aligner is, like for the case of other omics tools, not straightforward. Fundamentally, it needs to be evaluated in the context of the downstream application and the target goals. There is no hard and fast rule to determine the best performing tool.

Some organized efforts to evaluate aligners have been the Alignathon [190], and the RGASP for RNA-seq data [191]. These benchmarking exercises help in defining guidelines, and identifying metrics and datasets, and are great collaborative efforts for independent assessment. However, the inherent detail that needs to be taken into account in analyzing and interpreting the results from from all aligners muddles possible recommendations. What is more, the fast-paced technological advances, and the perpetual development of computational tools make the benchmarks to not age well, despite the tremendous effort behind them.

5.5 Selection of a reference-based aligner

For our purpose, it is clear that a prospect aligner to study is constrained by the necessity to use quality score metadata for the mapping of sequences. In our experience, identifying the usage of quality scores, and what is more, how their values are used, in the methods for aligning short read sequences, is not an information upfront to find. However, we have learned of some resources that attempt to survey aligners comprehensively, to the extent possible, and widely through a considerable number of features [188]. Far from attempting to benchmark tools, this work is a compendium of mappers classified by several features for quick comparison. The practitioner can discover the tools and rapidly make a first assessment to pick those which are suitable for their goals. The compendium was originally envisioned to be regularly updated and a work in progress; unfortunately the access to the list is not longer available but its content is referenced in the manuscript [188]. In this line, perhaps other open collaborative efforts[2] could be maintained more fruitfully.

[2] https://en.wikipedia.org/wiki/List_of_sequence_alignment_software

Albeit somewhat dated now, the compendium shows trends in the awareness of quality scores, a feature that is moderately exploited to align DNA and RNA-seq data.

The design of current alignment methods are subject to two fundamental considerations that come as a result of the improvement in sequencing technologies and experimental protocols. First, aligners need to be optimized for speed and memory usage to cope with the increasing sequencing capacities, and second, they need to adapt and address the error profiles in reads produced by different sequencing platforms [192]. Therefore, the algorithms need to be efficient and map with high accuracy, a tradeoff that is handled differently by each tool. It is often unclear, however, how this tradeoff impacts the overall performance of a mapping technique.

The work in [193] made us gain a better understanding of how algorithmic features and different input properties (the type of reference genome and read length) play out in alignment performance. We have found in benchmarking studies that idiosyncratic qualities of alignment tools are usually 'homogenized', in the sense that they start with a 'common ground' for comparison, in that each tool adheres to their own set of initial conditions: their default configuration parameters. In fact, it has been reported than more than half of benchmarking studies use tools with default parameter settings [48]. Then, for consistent evaluation, the metrics of the benchmark are applied uniformly, without regard of particular features and characteristics of each tool. With this procedure, the comparison is simplified at the expense of clarity of analysis. As the consensus of aligned reads tends to be large between aligners, it is the small variations in mapping results (usually in the order of units to tenths of a percent, but it depends on the choice of the benchmark) that make the difference in alignment performance. These capricious changes result from the particularities of each algorithm. Accordingly, the devil is in the details. The study in [193] wades through different features supported by the aligners under analysis, like the use of seeds, allowance of indels, use of quality scores, etc. The default options of the tested tools, like the number of permitted mismatches in the seed and read, the seed length, quality threshold, etc, are also considered. The evaluation criteria looked at throughput and mapping percentage, both function of the above elements, to determine

the performance of the aligners. They experimented with mapping options and configured consistently the aligners, such that all of them would run under the same circumstances. For example, with a specific number of mismatches allowed and corresponding quality threshold, they found that differences in mapping rates where caused by (i) the default configuration options for some tools; (ii) incorrect alignments that increased the mapping percentage for some other tools; and (iii) the lack of support for the feature by the tool, which for this particular condition resulted in a better mapping percentage. Another example was to examine the effect of seed length, with a fixed number of mismatches, on mapping percentages. For the tools that supported this feature, the expected result was observed: larger seeds limited the number of candidate alignment locations thus reducing the alignment percentage. One of the tools displayed this behavior but up to a point at which the opposite was observed, and the alignment percentage started to increase. This was due to a constraint in the backtracking search of the aligner, which ceased to look for candidate alignment locations rapidly, and concentrated instead on depth first searching over them.

The alignment process is affected by many factors. Each tool devises the tradeoff between speed and quality that yields the aligner's particular performance. Configuring and exploiting the features of an aligner is important to achieve good performance, and in a way is like continually adjusting the balance of a seesaw board. It is up to the end user to identify first their needs and then match them with an appropriate mapping technique.

All things considered, we have decided on the aligner HISAT2 [164] to explore the effect of lossy quality scores. This aligner supports quality score metadata, and uses it for the computation of alignment scores, a primary requirement for our purpose. Built upon Bowtie2 [123], HISAT2 is in fact the evolution of this very well-know and popular aligner. Moreover, it has good adoption and performance [145, 194], it has stood the test of time and is open source; what is more, it is still been maintained[3]. In addition, it was designed to map both DNA and RNA-seq reads. Also, this choice is rather opportune since HISAT2 is the alignment tool that we have been using all along and throughout our investigation of omics applications.

[3] https://ccb.jhu.edu/software/hisat2/index.shtml

5.6 Lossy quality scores and alignment

The challenge to represent lossy quality scores in the alignment of sequence reads lies in maintaining the read's original alignment location(s) with the new simplified representation. In general, quality score values participate in the computation of suitable alignment locations for reads in quality-aware aligners. The way in which quality score values is factored in depends on the alignment technique, and their usage is not essential but clearly optional. Many aligners have been developed that do not rely on quality scores. This is readily noted in benchmark comparisons, which commonly include widely used aligners [187, 188, 145, 193, 195].

Using quality scores can improve alignment accuracy because the information they provide, the probability of error in the calling of each sequence base, can be incorporated to determine which positions in a read are more important to map [196, 187]. Quality scores can be used in very diverse ways among alignment tools, as the methods prioritize this metadata rather differently.

For example, one of the most widely used reference-based aligners, BWA [197, 121], incorporates quality scores in a measure for the reliability of alignments. The aligner does this by defining a mapping quality score that represents the error probability of each read alignment. Quality scores are not used in BWA's alignment algorithm but rather they are used to support alignment results. Moreover, this score is used by the aligner to estimate the insert size distribution in paired-end mapped reads.

In constrast, quality scores can be incorporated at the core of an aligner's algorithm to guide the alignment decision. This is the case for Novoalign [198], another very well-known reference-based aligner, which consistently ranks well in alignment accuracy. Novoalign uses quality score information in its penalization system to score candidate alignment locations for each input sequence read.

Our purpose is to investigate the contribution of quality scores to alignment in HISAT2. In other words, we are interested to determine their relevance as per how their inclusion con-

tributes to aligning sequences. The role of quality scores in alignment is framed within HISAT's scoring system, and understanding it will be the way through finding a simplified representation for the quality scores that circumvents undesirable effects on alignment. Concretely, the goal is to preserve alignment locations as if no modification to the values of quality scores was done before alignment, that is, we aim at varying the quality scores transparently.

We ask, under what circumstances quality score values are, or become, informative for determining the alignment location of a read? To address this question we look into how quality scores weigh in on HISAT2's scoring system. Precisely, we focus on the following points:

- Quality scores partake in the computation of a measure for every read, called Alignment Score (AS), whose value is used by the aligner to classify reads as aligned or unaligned; reads that satisfy their alignment score are said to be aligned. Thus, the AS can be seen as a proxy to measure the effect quality scores have on alignment. But how are alignment scores affected by quality scores?

- Compressors of quality scores simplify their representation to reduce entropy and gain storage space. These lossy representations are deliberate changes to quality score values. But how are alignment scores affected by the intentional modification of quality scores?

- The above points guide our investigation into looking how to keep alignment scores unaffected, while intentionally changing the values of quality scores. In achieving this, the original alignment locations for the reads are preserved.

We address the first point by studying the alignment score's penalty function, and show when and how quality scores contribute to their value. Then, we explore the effect of quality score compression on alignment scores, and on the alignment location of reads. Finally, we find how to achieve alignment score invariance, and thus preserve sequence alignment, by transparently representing lossy quality scores.

5.7 Alignment score and quality scores: Penalization scores

Aligning sequences consists in lining up characters to reveal similarity. However, the aligner cannot always assign a read to its point of origin with high confidence, thus it makes an educated guess about its origin in the reference sequence.

HISAT2 quantifies how similar the sequence of a read is to the reference sequence it aligns to by computing an alignment score for the read. The aligner starts with the assumption that no difference exists between the read sequence `r` and the segment of the reference sequence `R`, pointed to by the alignment location, it aligns to. If this condition is satisfied, the best possible alignment score is assigned to the read, which is zero. This the largest, non-negative value the alignment score can take. The concept of alignment score does not apply to unaligned reads, as such HISAT2 does not report a value, nor the AS metric for these reads. As dissimilarities are found between `r` and `R`, HISAT2 penalizes each discrepant sequence character. Penalty values are always negative and are added together to compute the total alignment score for the read `r`. For an alignment to be considered good enough, or valid, it must have an alignment score with a value no less than the minimum score threshold `t`. The threshold is configurable and a function of the read length `x`, and its default value is $t(x) = 0 - 0.2 \times (x)$. Thus, valid alignments meet or exceed the minimum score theshold and are capped at zero. For example, aligned sequences of 100 base-pairs long will have valid alignment scores in the range $-20 \leq AS \leq 0$.

There are four types of penalizations, and each is scored differently:

- Ambiguous characters (N). The penalty is set in positions where the read, reference or both, contain an ambiguous character such as N. For each ambiguous character the penalization is 1

- Gaps. Affine gaps in the read or the reference are penalized for their occurrence (gap opening, `O`), and for each position they span (gap extension `E`). The sum of both values defines the penalization for the gap. The penalty for a read gap of length `n` is $O + n \times E$, and its default is $5 + n \times 3$. The same expression applies for a reference gap of length `n`

- Soft-clips (sc). Reads can be aligned in a way such that they are trimmed at one or both extremes, because some of the characters at their ends do not match the reference. Omitted characters are trimmed or soft-clipped from the read to produce a valid alignment. Each character that is soft-clipped receives a penalty value defined by the penalty function

$$\mathtt{P} = \mathtt{MN} + \left\lfloor (\mathtt{MX} - \mathtt{MN}) \, \frac{\mathtt{min}(\mathtt{Q}, 40)}{40} \right\rfloor \tag{5.1}$$

  where $\mathtt{MX} = 2$, and $\mathtt{MN} = 1$ are the default values, and $\mathtt{Q}$ is the quality score for the soft-clipped character

- Mismatches (mm). These are discrepant characters between the read and the reference. Each mismatch is penalized using the penalty function for soft-clips. However, the parameters values change for mismatches, and they default to $\mathtt{MX} = 6$, and $\mathtt{MN} = 2$

Let's note that quality scores participate only in the penalization for mismatches and soft-clips. By solving the penalty function above for the full quality score scale, and for both mismatches and soft-clips, we get Table 5.1.

To get a first estimate of the occurrence of ambiguous characters, gap openings and gap extensions, and mismatches in a sequence file, we created synthetic RNA-seq data of the human chromosome X with the Flux simulator, and aligned it with HISAT2. We decided to work with sets of one thousand reads to make the processing and parsing of aligned files manageable. Then, we randomly sampled ten instances of a thousand aligned reads each, and parsed out the information for N (XN), gaps (XO and XG), mismatches (XM), and the alignment score (AS) from the aligned SAM file. For each instance, we plotted the respective median value and computed also the average of all instances. The graph is shown in Figure 5.1.

The type of penalization missing in Figure 5.1 is the soft-clipping. Soft-clips, in comparison to other features, are not reported as a direct value in the aligned file. Instead, soft-clipped

Table 5.1 – Penalty values for mismatches and soft-clips. Penalties are always negative values. P_{mm}: Penalization for mismatches; P_{sc}: Penalization for soft-clips.

ASCII	Q	$f = \frac{min(Q,40)}{40}$	$P_{mm} = -(2 + \lfloor 4 \times f \rfloor)$	$P_{sc} = -(1 + \lfloor 1 \times f \rfloor)$
73	40	1	-6	-2
72	39	0.975	-5	-1
71	38	0.95		
70	37	0.925		
69	36	0.9		
68	35	0.875		
67	34	0.85		
66	33	0.825		
65	32	0.8		
64	31	0.775		
63	30	0.75		
62	29	0.725	-4	
61	28	0.7		
60	27	0.675		
59	26	0.65		
58	25	0.625		
57	24	0.6		
56	23	0.575		
55	22	0.55		
54	21	0.525		
53	20	0.5		
52	19	0.475	-3	
51	18	0.45		
50	17	0.425		
49	16	0.4		
48	15	0.375		
47	14	0.35		
46	13	0.325		
45	12	0.3		
44	11	0.275		
43	10	0.25		
42	9	0.225	-2	
41	8	0.2		
40	7	0.175		
39	6	0.15		
38	5	0.125		
37	4	0.1		
36	3	0.075		
35	2	0.05		
34	1	0.025		
33	0	0		

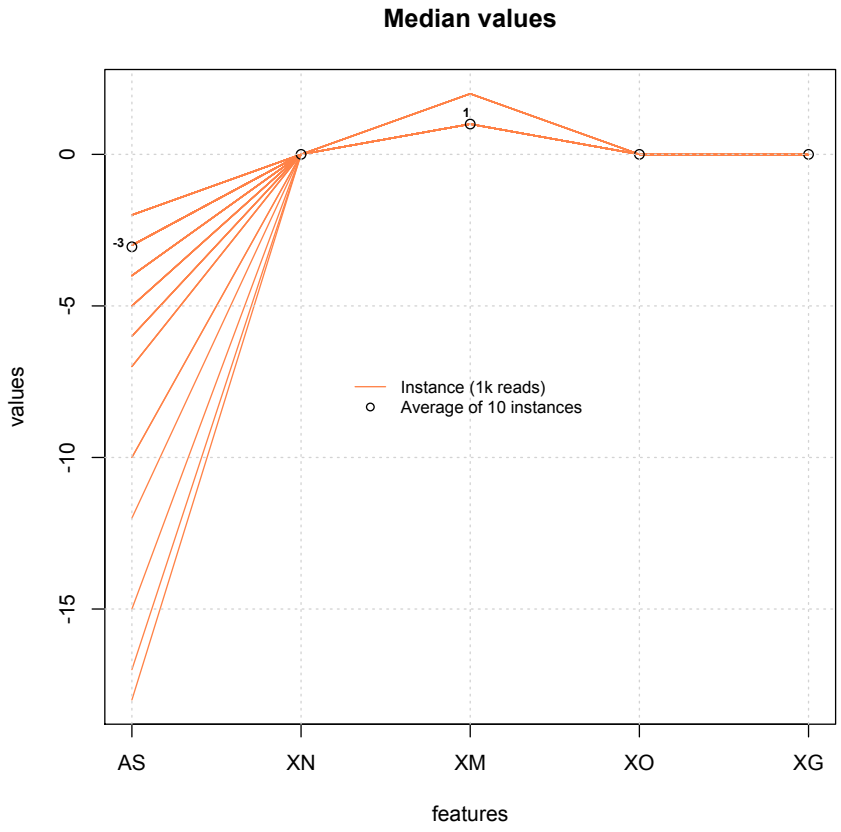


Figure 5.1 – Occurrence of ambiguous characters (XN), gap openings (XO) and gap extensions (XG), and mismatches (XM) in a collection of 1k reads from a simulated sequence file. In this example, all reads aligned without gaps or ambiguous characters, and with an average of one mismatch. The value for the alignment score spans across the range $-20 \leq \texttt{AS} \leq 0$.

characters are detailed as a string, which points to their occurrences in the aligned read. Complementing the information in Figure 5.1 with soft-clip events that occur in the same collection of reads, we get the distribution of penalty features shown in the pie chart in Figure 5.2. Roughly one third of reads are unaligned, and 13% of reads are penalized.

However, Figure 5.2 is misleading in the sense that, in the general case, mismatches and soft-clips can occur simultaneously in the aligned read, and their distribution cannot be neatly separated as in the pie chart in Figure 5.2. Nevertheless, the effect of quality scores on AS can in principle be separated by mismatches and soft-clips.

Looking at Table 5.1, the set of penalty values for mismatches are $S_{\texttt{mm}} = \{-2, -3, -4, -5, -6\}$.

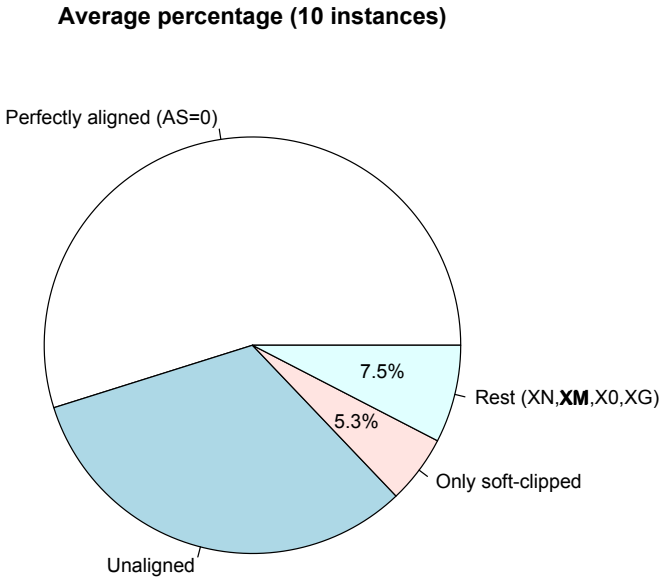


Figure 5.2 – Distribution of penalty features in a 10k read simulated file. For this file, only two types of penalties were reported in the aligned file, soft-clips and mismatches. The proportion of reads whose quality scores are unused because they are either perfectly aligned or unaligned is 90%.

Valid alignments with mismatches for reads of 100 base-pairs have an $-20 \leq \texttt{AS} \leq -2$. The maximum number of mismatches for a valid alignment is 10, because the alignment threshold is `-20`, which is reached for the worst case with ten mismatches of value `-2` each. One mismatch is the minimum number possible, whatever its value. To observe how the penalties for mismatches impact AS, we computed all combinations of set $\texttt{S}_{\texttt{mm}}$ up to 10 elements, the maximum number of mismatches for a valid alignment. More precisely, we computed the combinations $\texttt{C}(\ 5, \texttt{x}\ )$ for $1 \leq \texttt{x} \leq 10$. The sum of the elements in each tuple of each combination is the alignment score for that tuple. Refer to Figure 5.3.

As we are interested in observing the effect of QS in the computation of the AS for mismatches, however they may happen, we focused not on the organization of each tuple but rather in grouping tuples that yield the same alignment score. Tuples grouped by their AS are shown in Figure 5.4; those with identical AS align horizontally. In the figure, each circle represents a tuple whose AS is determined by adding up their elements. For example, the tuples $(-2, -4)$

1	C(5,1)	C(5,2)	C(5,3)	C(5,4)	C(5,5)	C(5,6)	C(5,7)	C(5,8)	C(5,9)	C(5,10)
2	#N=5	#N=15	#N =35	#N=70	#N=126	#N=210	#N=330	#N=495	#N=715	#N=1001
3	2	2,2	2,2,2	2,2,2,2	2,2,2,2,2	2,2,2,2,2,2	2,2,2,2,2,2,2	2,2,2,2,2,2,2,2	2,2,2,2,2,2,2,2,2	2,2,2,2,2,2,2,2,2,2
4	3	2,3	2,2,3	2,2,2,3	3,3,3,3,3	3,3,3,3,3,3	3,3,3,3,3,3,3	3,3,3,3,3,3,3,3	3,3,3,3,3,3,3,3,3	3,3,3,3,3,3,3,3,3,3
5	4	2,4	2,2,4	2,2,2,4	4,4,4,4,4	4,4,4,4,4,4	4,4,4,4,4,4,4	4,4,4,4,4,4,4,4	4,4,4,4,4,4,4,4,4	4,4,4,4,4,4,4,4,4,4
6	5	2,5	2,2,5	2,2,2,5	5,5,5,5,5	5,5,5,5,5,5	5,5,5,5,5,5,5	5,5,5,5,5,5,5,5	5,5,5,5,5,5,5,5,5	5,5,5,5,5,5,5,5,5,5
7	6	2,6	2,2,6	2,2,2,6	6,6,6,6,6	6,6,6,6,6,6	6,6,6,6,6,6,6	6,6,6,6,6,6,6,6	6,6,6,6,6,6,6,6,6	6,6,6,6,6,6,6,6,6,6
8		3,3	2,3,3	2,2,3,3	2,2,2,2,3	2,2,2,2,2,3	2,2,2,2,2,2,3	2,2,2,2,2,2,2,3	2,2,2,2,2,2,2,2,3	2,2,2,2,2,2,2,2,2,3
9		3,4	2,3,4	2,2,3,4	2,2,2,2,4	2,2,2,2,2,4	2,2,2,2,2,2,4	2,2,2,2,2,2,2,4	2,2,2,2,2,2,2,2,4	2,2,2,2,2,2,2,2,2,4
10		3,5	2,3,5	2,2,3,5	2,2,2,2,5	2,2,2,2,2,5	2,2,2,2,2,2,5	2,2,2,2,2,2,2,5	2,2,2,2,2,2,2,2,5	2,2,2,2,2,2,2,2,2,5
11		3,6	2,3,6	2,2,3,6	2,2,2,2,6	2,2,2,2,2,6	2,2,2,2,2,2,6	2,2,2,2,2,2,2,6	2,2,2,2,2,2,2,2,6	2,2,2,2,2,2,2,2,2,6
12		4,4	2,4,4	2,2,4,4	2,3,3,3,3	2,3,3,3,3,3	2,3,3,3,3,3,3	2,3,3,3,3,3,3,3	2,3,3,3,3,3,3,3,3	2,3,3,3,3,3,3,3,3,3
13		4,5	2,4,5	2,2,4,5	2,4,4,4,4	2,4,4,4,4,4	2,4,4,4,4,4,4	2,4,4,4,4,4,4,4	2,4,4,4,4,4,4,4,4	2,4,4,4,4,4,4,4,4,4
14		4,6	2,4,6	2,2,4,6	2,5,5,5,5	2,5,5,5,5,5	2,5,5,5,5,5,5	2,5,5,5,5,5,5,5	2,5,5,5,5,5,5,5,5	2,5,5,5,5,5,5,5,5,5
15		5,5	2,5,5	2,2,5,5	2,6,6,6,6	2,6,6,6,6,6	2,6,6,6,6,6,6	2,6,6,6,6,6,6,6	2,6,6,6,6,6,6,6,6	2,6,6,6,6,6,6,6,6,6
16		5,6	2,5,6	2,2,5,6	3,3,3,3,4	3,3,3,3,3,4	3,3,3,3,3,3,4	3,3,3,3,3,3,3,4	3,3,3,3,3,3,3,3,4	3,3,3,3,3,3,3,3,3,4
17		6,6	2,6,6	2,2,6,6	3,3,3,3,5	3,3,3,3,3,5	3,3,3,3,3,3,5	3,3,3,3,3,3,3,5	3,3,3,3,3,3,3,3,5	3,3,3,3,3,3,3,3,3,5
18			3,3,3	2,3,3,3	3,3,3,3,6	3,3,3,3,3,6	3,3,3,3,3,3,6	3,3,3,3,3,3,3,6	3,3,3,3,3,3,3,3,6	3,3,3,3,3,3,3,3,3,6
19			3,3,4	2,3,3,4	3,4,4,4,4	3,4,4,4,4,4	3,4,4,4,4,4,4	3,4,4,4,4,4,4,4	3,4,4,4,4,4,4,4,4	3,4,4,4,4,4,4,4,4,4
20			3,3,5	2,3,3,5	3,5,5,5,5	3,5,5,5,5,5	3,5,5,5,5,5,5	3,5,5,5,5,5,5,5	3,5,5,5,5,5,5,5,5	3,5,5,5,5,5,5,5,5,5
21			3,3,6	2,3,3,6	3,6,6,6,6	3,6,6,6,6,6	3,6,6,6,6,6,6	3,6,6,6,6,6,6,6	3,6,6,6,6,6,6,6,6	3,6,6,6,6,6,6,6,6,6
22			3,4,4	2,3,4,4	4,4,4,4,5	4,4,4,4,4,5	4,4,4,4,4,4,5	4,4,4,4,4,4,4,5	4,4,4,4,4,4,4,4,5	4,4,4,4,4,4,4,4,4,5
23			3,4,5	2,3,4,5	4,4,4,4,6	4,4,4,4,4,6	4,4,4,4,4,4,6	4,4,4,4,4,4,4,6	4,4,4,4,4,4,4,4,6	4,4,4,4,4,4,4,4,4,6
24			3,4,6	2,3,4,6	4,5,5,5,5	4,5,5,5,5,5	4,5,5,5,5,5,5	4,5,5,5,5,5,5,5	4,5,5,5,5,5,5,5,5	4,5,5,5,5,5,5,5,5,5
25			3,5,5	2,3,5,5	4,6,6,6,6	4,6,6,6,6,6	4,6,6,6,6,6,6	4,6,6,6,6,6,6,6	4,6,6,6,6,6,6,6,6	4,6,6,6,6,6,6,6,6,6
26			3,5,6	2,3,5,6	5,5,5,5,6	5,5,5,5,5,6	5,5,5,5,5,5,6	5,5,5,5,5,5,5,6	5,5,5,5,5,5,5,5,6	5,5,5,5,5,5,5,5,5,6
27			3,6,6	2,3,6,6	5,6,6,6,6	5,6,6,6,6,6	5,6,6,6,6,6,6	5,6,6,6,6,6,6,6	5,6,6,6,6,6,6,6,6	5,6,6,6,6,6,6,6,6,6
28			4,4,4	2,4,4,4	2,2,2,3,3	2,2,2,2,3,3	2,2,2,2,2,3,3	2,2,2,2,2,2,3,3	2,2,2,2,2,2,2,3,3	2,2,2,2,2,2,2,2,3,3
29			4,4,5	2,4,4,5	2,2,2,4,4	2,2,2,2,4,4	2,2,2,2,2,4,4	2,2,2,2,2,2,4,4	2,2,2,2,2,2,2,4,4	2,2,2,2,2,2,2,2,4,4
30			4,4,6	2,4,4,6	2,2,2,5,5	2,2,2,2,5,5	2,2,2,2,2,5,5	2,2,2,2,2,2,5,5	2,2,2,2,2,2,2,5,5	2,2,2,2,2,2,2,2,5,5

Figure 5.3 – Tuples for mismatches penalty values. The first thirty elements in the list of tuples for all combinations is shown. N is the number of tuples for the particular combination.

and $(-2,-2,-2)$ have the same $\texttt{AS} = -6$, and are aligned horizontally in the graph. We note how most ways in which mismatches can occur exceed the alignment score threshold. This means that the search for valid alignments, those above the alignment threshold, can be reduced substantially to tuples with ten or less elements with an $\texttt{AS} \geq -20$.

We followed a similar procedure for soft-clips to find the combinations of the elements in the set $\texttt{S}_{\texttt{sc}} = \{-1,-2\}$ (refer to Table 5.1). We computed the combinations $\texttt{C}(2,\texttt{x})$ for $1 \leq \texttt{x} \leq 20$ and calculated their respective alignment score. The worst case for a valid alignment with sof-clips is a tuple with 20 elements, each with a value `-1`. The graph in Figure 5.5 shows the alignment score as a function soft-clips, which are organized in tuples according to the possible combinations of their penalty values in set $\texttt{S}_{\texttt{sc}}$.

5.8 Alignment score, alignment locations and lossy quality score compressors

Ultimately, the effect in changing the representation of quality scores in alignment is the repercussion on the assignment of locations of aligned reads. The assignment is subordinate

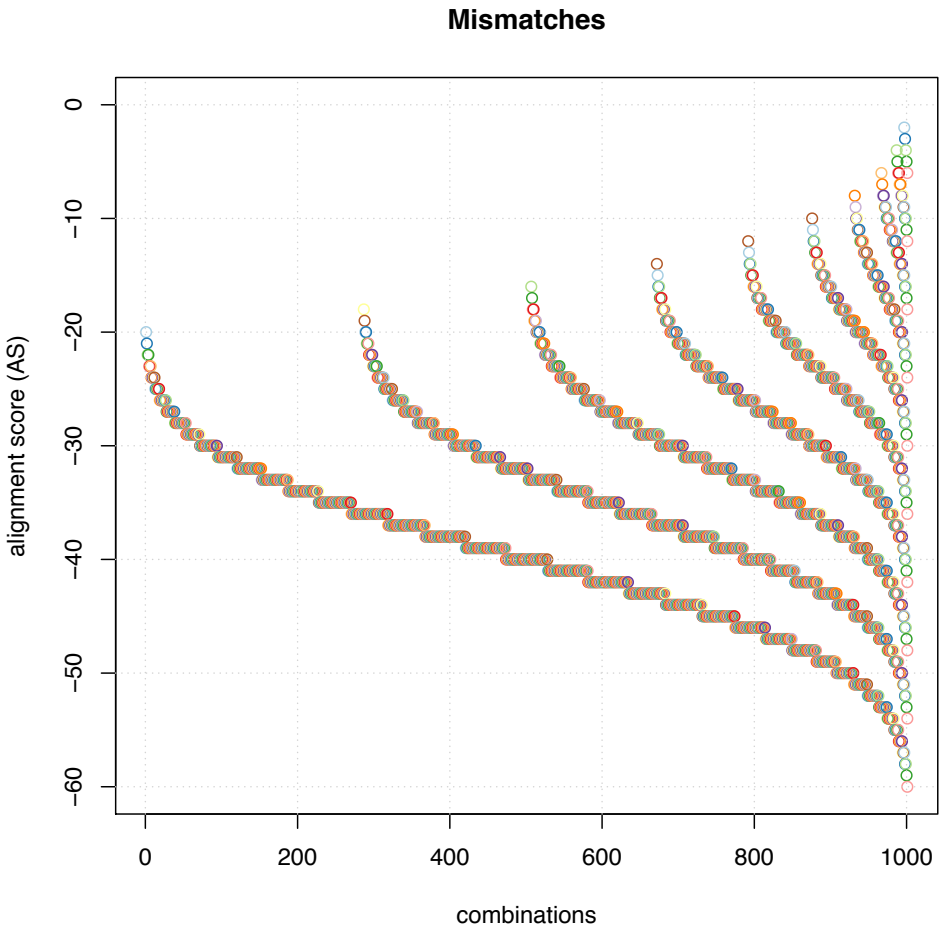


Figure 5.4 – Alignment score and penalty values for mismatches.

to the alignment score, which acknowledges its validity, and in studying how compressors disrupt alignment scores we can gain understanding of how they modify quality scores and thus observe their significance for alignment. We now explore how alignment scores, and alignment locations, are affected by deliberate changes to the quality scores of sequence reads. Lossy compressors of quality scores vary greatly in their methods for changing quality score representation, and interestingly, the impact caused by these tools on variant calling and gene expression is rather similar. Instead of focusing on the particularities of each tool, given than all are effective at marginally affecting downstream results, we attempt to get a grasp of how their methods influence changes in the assignment of alignment locations.

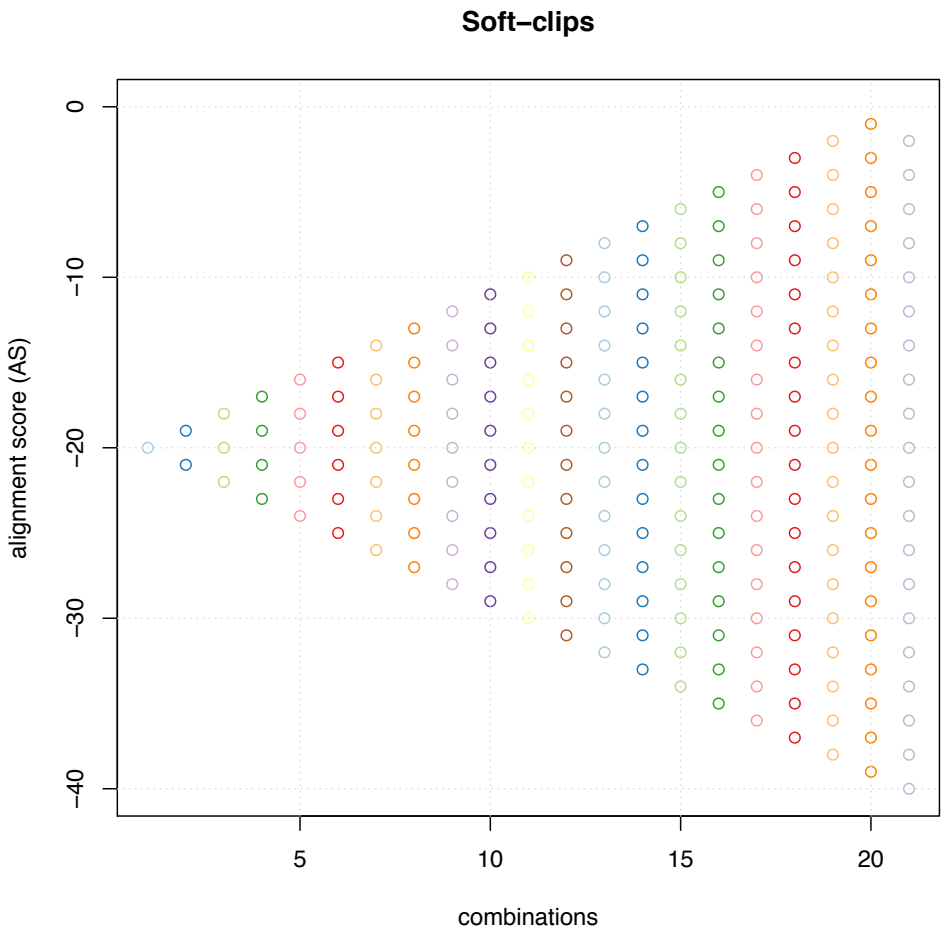


Figure 5.5 – Alignment score and penalty values for soft-clips.

Lossy compressors can modify quality scores such as to model future occurrences of their values in the sequence [85], quantize uniformly in blocks based on values to suggested metrics [88] or use corpuses of sequences to instruct the location of quality scores to keep [87]. We cannot know before alignment how the unaligned sequences will be described in terms of mismatches and soft-clips (the features whose penalty values are subject to quality scores), but at the level of detail of sequence characters, we can observe how different methods approach the modification of quality score values.

We simulated ten thousand, 100 base-pair length, RNA-seq reads from the human chromosome X, and ordered them by read name. Then, the representation to their quality scores

was changed with three lossy compression tools: quartz [87], qvz [85], and prblock [88]. We took the first 100 reads out of each output file, and compared them graphically. The new representation to the quality scores given by these tools is shown in Figure 5.6. Each horizontal line depicts the quality score sequence for a read (100 quality score values), and because the file is ordered by read name, horizontal lines across compressors describe the same sequence of quality scores. Changes to the quality score values are marked in yellow. Untouched quality scores are in red.

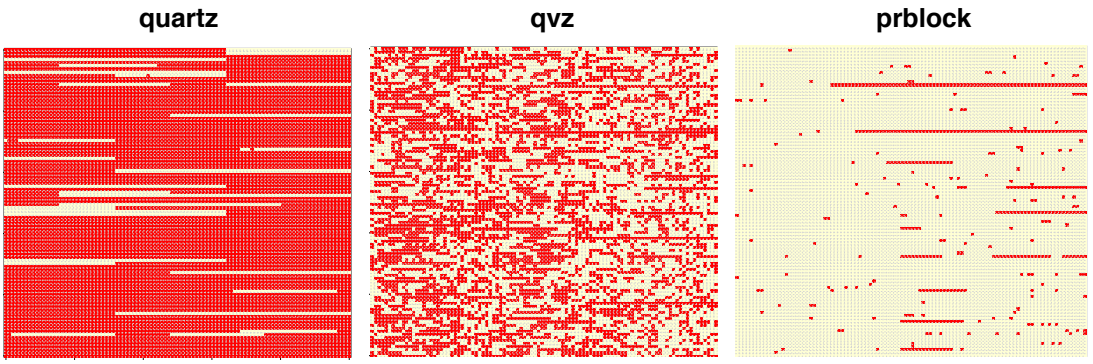


Figure 5.6 – Blueprint of quality score changes by different lossy compressors. Each horizontal line represents a sequence of quality scores whose values have been modified (yellow) or not (red) by the lossy compression tool. Horizontal lines across the three tools depict the same quality score sequence.

The parameters of each tool were adjusted across them such as they would output the same level of compression. Thus, the visualization in Figure 5.6 blueprints different approaches to modifying quality score values for the same compression rate.

We hypothesized the existence of a degradation level to the quality scores such that, despite changes to alignment scores, the assignment of alignment positions get to be mostly preserved. In our simulations, mismatches and soft-clips sparsely occurred in around 10% of aligned sequences. Thus, aggressive compression can be performed pre-alignment, and perhaps even blindly with little impact. Therefore, we looked into test different levels of compression to observe changes in both AS and alignment locations. We carried this out in four steps.

5.8.1 Step 1: Generate input data

First, we created synthetic sequences to have the reference location origins for each read. We ran dozens of simulations to generate synthetic data from human chromosomes X and Y, and for chromosome 1. Unaligned data in the form of FASTQ files, and a reference file in the form BED files[4], were the output from the simulator. The reference BED file contains information on the origin location, among other things, for all reads in a simulated FASTQ file.

The sequences were then aligned with HISAT2 to produce SAM files. See Figure 5.7. Reads in each file were sampled randomly and collected in groups of one thousand. Three samples of one thousand reads each were collected for each simulated FASTQ file, and the same was done on their corresponding aligned files.

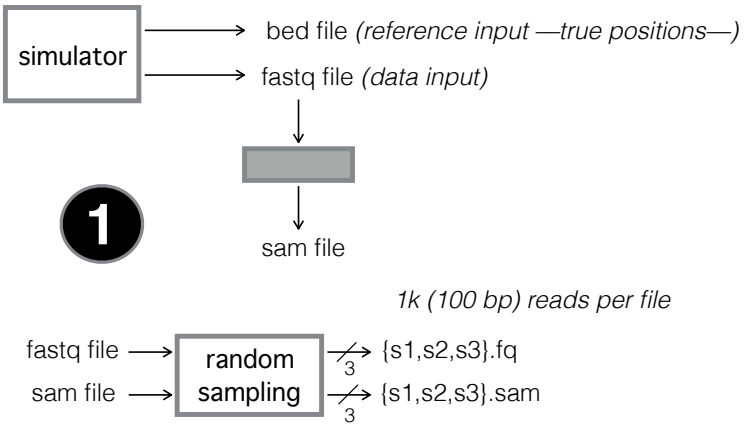


Figure 5.7 – Preparation of input data for the experiments.

5.8.2 Step 2: Apply lossy compresion to quality scores

Second, the three compressors used previously were configured to match compression levels. The discovery of the appropriate parameters for this task was manually done by trial and error for each input file, as the tools expect as entry point the value to their parameters, not a compression rate. As such, the distortion level is approximate across two tools, qvz

[4] https://useast.ensembl.org/info/website/upload/bed.html

and prblock. As for the third tool, quartz, the configuration was not possible because the compression rate is fixed and harcoded.

For each FASTQ sample five compression levels were tested, and are shown in Figure 5.8. Compression levels are measured in bits per quality score. The outputs are also FASTQ files but with a different representation to the quality score sequences. We note that prblock extracts the quality scores from an aligned file but the tool does not use alignment information for compression.

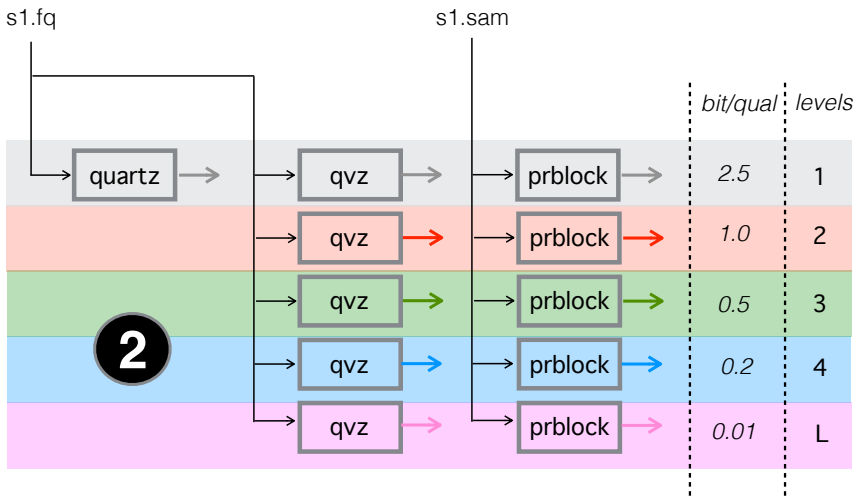


Figure 5.8 – Levels of compression tested in the experiments.

5.8.3 Step 3: Run comparisons and generate output tables

Third, input FASTQ files with and without distortion to their quality scores were aligned. Aligned reads were intersected with the reference BED file to extract the relevant locations to the subset of reads in the sample FASTQ file. The alignment location for each read was compared to the 'ground truth' location reported for that read in the BED file, and those reads whose locations matched were filtered out. Following a similar procedure, the corresponding alignment scores were also extracted.

Figure 5.9 depicts this process. The output is a set of two tables, one focused on alignment positions, and the other on alignment scores.

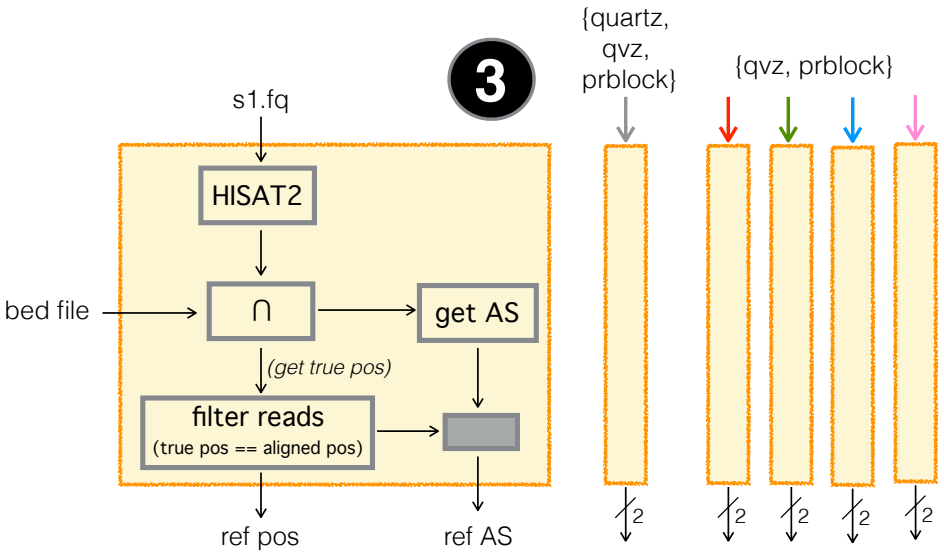


Figure 5.9 – Process to construct tables of alignment positions and alignment scores for each input file.

5.8.4 Step 4: Analisis of results

Lastly, the output tables from last step are organized for each compression level and for each compression tool. The table for alignment positions reports the positions found by HISAT2 under each compression level, and for each read. The reference positions from the BED file are also included. Similarly, the table for alignment scores reports their values for each read and compression level. The alignment score for the undistorted FASTQ file is also reported, and it is considered as the reference (or 'true') value. See Figure 5.10. A table for positions and AS are generated for each input FASTQ file.

5.8.5 Experimentation

An example of the output of one experiment is shown in Figures 5.11 and 5.12; only a portion of the tables are shown. The columns that report the effect of each compression level on alignment locations and alignment scores are organized and labeled by increasing level of compression. For example, qvz applied at compression level 1 (2.5 bits per quality score) is reported in the column qvz1, and more aggressive compression levels are reported in columns

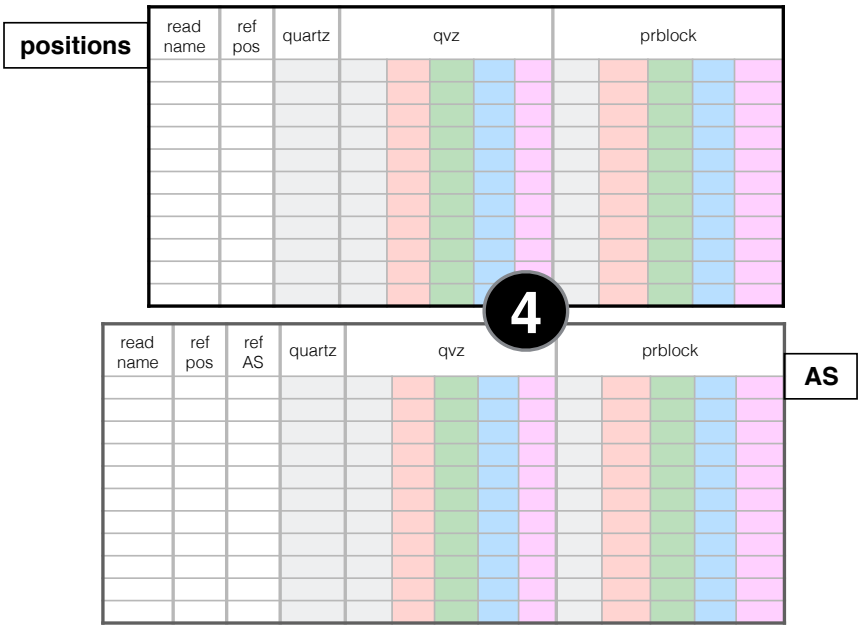


Figure 5.10 – Output of one experiment: tables for alignment positions and alignment scores.

to its right.

For each sample we compared the alignment location of their reads post-compression against corresponding alignment locations in the original, aligned but uncompressed, FASTQ file. We did this to discover the distribution of reads whose alignment location changed as a consequence of quality score compression.

Figure 5.13 depicts the procedure with the actual result (blue text) for an input sample of one thousand reads, simulated from chromosome X. First, each uncompressed FASTQ sample is processed to produce a reference SAM file (refer to the blue block in Figure 5.13).

The file is parsed, and aligned reads are classified by alignment score in two sets:

(i) Reads that aligned perfectly such that the aligned read sequence is identical to the sequence it aligned to

(ii) Reads that aligned with some errors, and therefore their `AS` < 0

The uncompressed FASTQ file is then fed to the three compression tools to generate the table

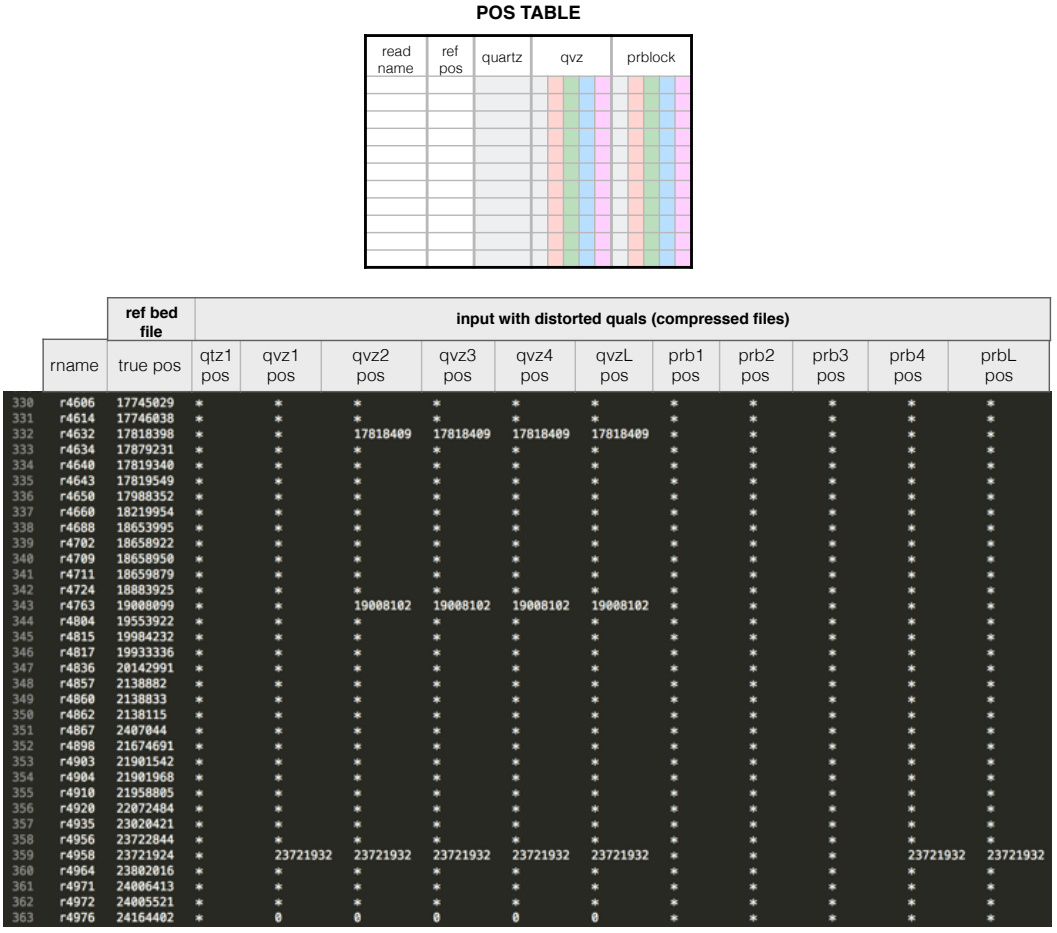
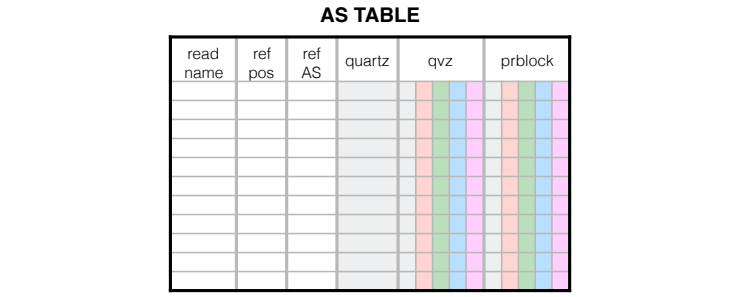


Figure 5.11 – Example of output table for alignment locations. Starred entries indicate the situations where compression did not affect the assignment of the alignment location for the read. Numeric entries under the columns for different compression levels indicate a change in the alignment position for the read, that is, it is the report for the new alignment location found by HISAT2 for that read.

for alignment locations described above. Refer to Figure 5.11.

The alignment locations obtained post-compression from the table are compared to corresponding alignment locations in the original file, which have already been classified by AS; see Figure 5.13. The result is two sets of reads reads grouped by whether their alignment score is zero or negative. Further, each set groups reads whose alignment location changed, or not, after quality score compression.

Results from the workflow in Figure 5.13 are reported in two tables for each tested sample. A



	rname	true AS	qtz1 AS	qvz1 AS	qvz2 AS	qvz3 AS	qvz4 AS	qvzL AS	prb1 AS	prb2 AS	prb3 AS	prb4 AS	prbL AS
330	r4606	0	0	0	0	0	0	0	0	0	0	0	0
331	r4614	0	0	0	0	0	0	0	0	0	0	0	0
332	r4632	-6	-6	-6	-11	-11	-11	-11	-6	-6	-6	-6	-6
333	r4634	0	0	0	0	0	0	0	0	0	0	0	0
334	r4640	0	0	0	0	0	0	0	0	0	0	0	0
335	r4643	0	0	0	0	0	0	0	0	0	0	0	0
336	r4650	0	0	0	0	0	0	0	0	0	0	0	0
337	r4660	0	0	0	0	0	0	0	0	0	0	0	0
338	r4688	-5	-5	-6	-10	-9	-10	-10	-5	-5	-5	-6	-6
339	r4702	0	0	0	0	0	0	0	0	0	0	0	0
340	r4709	0	0	0	0	0	0	0	0	0	0	0	0
341	r4711	-6	-6	-6	-15	-15	-15	-15	-6	-6	-6	-6	-6
342	r4724	0	0	0	0	0	0	0	0	0	0	0	0
343	r4763	-2	-2	-3	-3	-3	-3	-3	-2	-2	-2	-2	-2
344	r4804	0	0	0	0	0	0	0	0	0	0	0	0
345	r4815	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5.12 – Example of output table for alignment scores. Alignment score values that were preserved after compression are marked with zeros, otherwise new alignment scores are reported.

portion of both tables is shown in Figure 5.14.

To observe how AS were affected in reads whose assignment of alignment location changed, we explored in further detail their values at different compression levels. An excerpt of this analysis is shown in Figure 5.15 for the reads in the red frame shown in Figure 5.14.

In our experiments we typically observed results like those shown in Figure 5.15, where changes to a couple of units in the AS led to changes in alignment locations. Largely more often, however, were changes of one unit to alignment scores. The pie chart in Figure 5.16 shows the distribution of alignment scores and alignment locations for a file of 100k simulated reads from the human chromosome X after running the workflow in Figure 5.13. For the distribution shown in the figure, the input file was compressed with prblock at a rate of 1 bit per quality score.

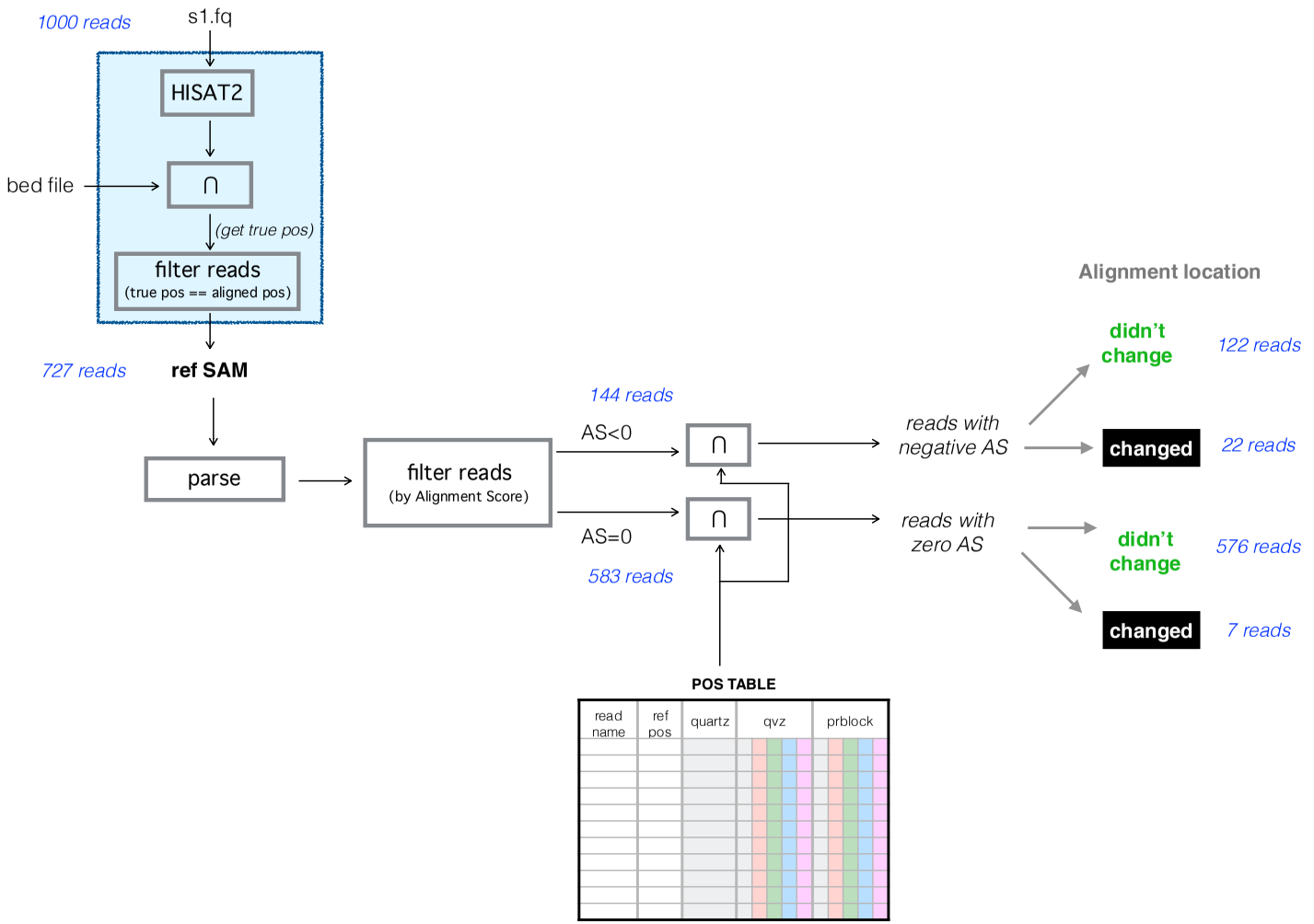


Figure 5.13 – Workflow to discover changes in the alignment location of reads subject to lossy compression of their quality scores.

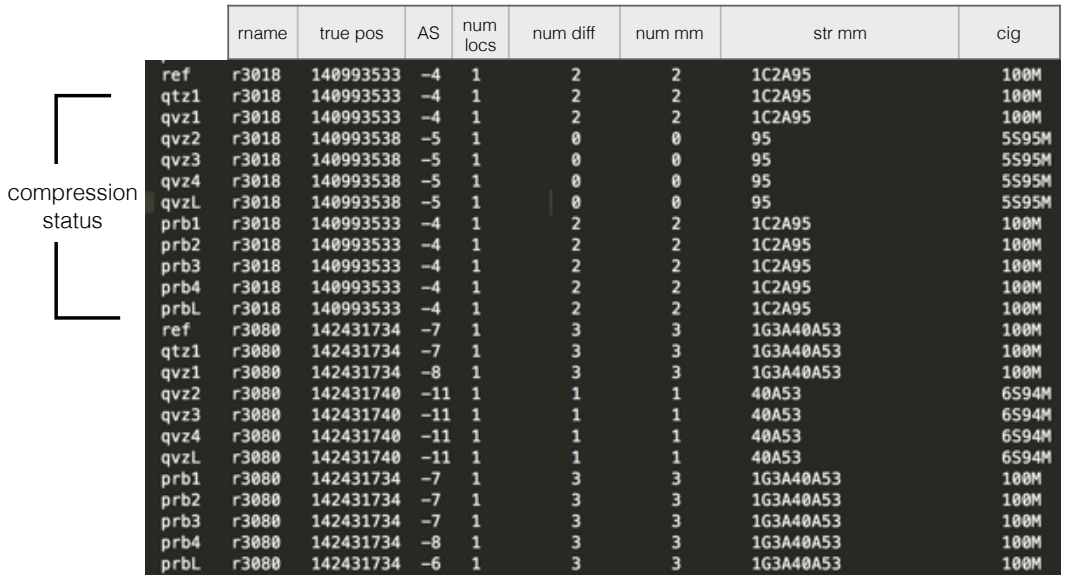
reads with AS = 0

	ref bed file		input with distorted quals (compressed files)											input without qual distortion (reference fastq file)					
	rname	true pos	qtz1 pos	qvz1 pos	qvz2 pos	qvz3 pos	qvz4 pos	qvzL pos	prb1 pos	prb2 pos	prb3 pos	prb4 pos	prbL pos	true AS	num locs	num diff	num mm	str mm	cig
145	r2476	134567796	*	*	*	*	*	*	*	*	*	*	*	0	1	0	0	100	100M
146	r2477	134598945	*	*	*	*	*	*	*	*	*	*	*	0	1	0	0	100	100M
147	r2509	134883626	*	*	*	134953757	134866352	*	134866352	134953757	134866352	134936498	134936498	0	5	0	0	100	100M
148	r2543	134992191	*	*	*	*	*	*	*	*	*	*	*	0	1	0	0	100	100M
149	r2546	135026973	*	*	*	*	*	*	*	*	*	*	*	0	1	0	0	100	100M
150	r2582	135290015	*	*	*	*	*	*	*	*	*	*	*	0	1	0	0	100	100M
151	r2587	135291423	*	*	*	*	*	*	*	*	*	*	*	0	1	0	0	100	100M
152	r2604	135299198	*	*	*	*	*	*	*	*	*	*	*	0	1	0	0	100	100M
153	r2610	135313908	*	*	*	*	*	*	*	*	*	*	*	0	1	0	0	100	100M

	ref bed file		input with distorted quals (compressed files)											input without qual distortion (reference fastq file)					
	rname	true pos	qtz1 pos	qvz1 pos	qvz2 pos	qvz3 pos	qvz4 pos	qvzL pos	prb1 pos	prb2 pos	prb3 pos	prb4 pos	prbL pos	true AS	num locs	num diff	num mm	str mm	cig
37	r2689	135482083	*	0	0	0	0	0	*	*	*	*	*	−20	1	10	10	60G0C0A0C0T3T0T0A0C0T27	100M
38	r2890	138664167	*	*	*	*	*	*	*	*	*	*	*	−11	1	0	0	90	90M10S
40	r3018	140993533	*	*	140993538	140993538	140993538	140993538	*	*	*	*	*	−4	1	2	2	1C2A95	100M
41	r3045	140996830	*	*	*	*	*	*	*	*	*	*	*	−3	1	1	1	30T69	100M
42	r3080	142431734	*	*	142431740	142431740	142431740	142431740	*	*	*	*	*	−7	1	3	3	1G3A40A53	100M
43	r3114	144901796	*	*	*	*	*	*	*	*	*	*	*	−15	1	0	0	90	90M10S
45	r3262	14883025	*	*	*	*	*	*	*	*	*	*	*	−4	1	1	1	77T22	100M
46	r3325	14910869	*	*	*	*	*	*	*	*	*	*	*	−2	1	1	1	22A77	100M
47	r3338	149014073	*	*	*	*	*	*	*	*	*	*	*	−2	1	1	1	64A35	100M
48	r3408	149935196	*	*	*	*	*	*	*	*	*	*	*	−10	1	1	1	64G31	96M4S
49	r3451	149936054	*	*	*	*	*	*	*	*	*	*	*	−4	1	1	1	70C29	100M
50	r3520	151122730	*	151122736	151122736	151122736	151122736	151122736	*	*	*	*	*	−6	1	3	3	0G2G1A94	100M
51	r3535	151122914	*	*	*	*	*	*	*	*	*	*	*	−6	1	3	3	92A2G0G3	100M
52	r3555	151824939	*	*	*	*	*	*	*	*	*	*	*	−1	1	0	0	99	99M1S
53	r3565	151876977	*	151928405	151928405	*	151928405	*	151928405	*	151928405	*	*	−3	2	1	1	31T68	100M
54	r3571	151900008	*	*	*	*	*	*	*	*	*	*	*	−4	1	2	2	95C0C3	100M
55	r3644	152140346	*	*	*	*	*	*	*	*	*	*	*	−1	1	0	0	99	99M1S
56	r3879	15301152	*	*	*	*	*	*	*	*	*	*	*	−15	1	0	0	86	14S86M

reads with AS < 0

Figure 5.14 – Output from the workflow shown in Figure 5.13. The tables show actual results from the same sample used in Figure 5.13. The red frame highlights two reads whose alignment positions changed after lossy compression; the reads held alignment scores of -4 and -7 before compression.



	rname	true pos	AS	num locs	num diff	num mm	str mm	cig
ref	r3018	140993533	-4	1	2	2	1C2A95	100M
qtz1	r3018	140993533	-4	1	2	2	1C2A95	100M
qvz1	r3018	140993533	-4	1	2	2	1C2A95	100M
qvz2	r3018	140993538	-5	1	0	0	95	5S95M
qvz3	r3018	140993538	-5	1	0	0	95	5S95M
qvz4	r3018	140993538	-5	1	0	0	95	5S95M
qvzL	r3018	140993538	-5	1	0	0	95	5S95M
prb1	r3018	140993533	-4	1	2	2	1C2A95	100M
prb2	r3018	140993533	-4	1	2	2	1C2A95	100M
prb3	r3018	140993533	-4	1	2	2	1C2A95	100M
prb4	r3018	140993533	-4	1	2	2	1C2A95	100M
prbL	r3018	140993533	-4	1	2	2	1C2A95	100M
ref	r3080	142431734	-7	1	3	3	1G3A40A53	100M
qtz1	r3080	142431734	-7	1	3	3	1G3A40A53	100M
qvz1	r3080	142431734	-8	1	3	3	1G3A40A53	100M
qvz2	r3080	142431740	-11	1	1	1	40A53	6S94M
qvz3	r3080	142431740	-11	1	1	1	40A53	6S94M
qvz4	r3080	142431740	-11	1	1	1	40A53	6S94M
qvzL	r3080	142431740	-11	1	1	1	40A53	6S94M
prb1	r3080	142431734	-7	1	3	3	1G3A40A53	100M
prb2	r3080	142431734	-7	1	3	3	1G3A40A53	100M
prb3	r3080	142431734	-7	1	3	3	1G3A40A53	100M
prb4	r3080	142431734	-8	1	3	3	1G3A40A53	100M
prbL	r3080	142431734	-6	1	3	3	1G3A40A53	100M

Figure 5.15 – Analysis of alignment scores for reads whose alignment location changed as a consequence of lossy compression to their quality scores. The column headers stand for the following: `num locs`, is the number of quality score characters modified as a consequence of lossy compression; `num diff`, is the number of differences between the aligned read sequence and the reference sequence; `num mm`, is the number of reported mismatches in the alignment of the read; `str mm`, is a character string describing the position mismatches in the alignment of the read; and cig, is a character string describing the number of soft-clips and their location. The strings of characters are directly reported in the SAM file.

We have consistently observed in our experiments that changes to the quality scores inevitably modify alignment, as per the report of reads whose alignment location changed post lossy compression. For all intents and purposes, the proportion of such changes to the overall proportion of unaffected reads looks awfully minor, and thus the impact is very small and capricious. However, as changes to alignment locations occur, expressing qualitatively the impact effect on alignment as being, generally, very small, appears vague and imprecise.

Therefore, instead of looking into "how far can we go in changing the representation of quality scores before changes in alignment locations occur?", we ask "how can we keep alignment locations unchanged while changing the representation of quality scores?". The answer is in achieving invariance for the alignment scores. That is, the goal is to avoid modifying alignment scores so as to preserve alignment locations. To do this, we need to know what not to affect in

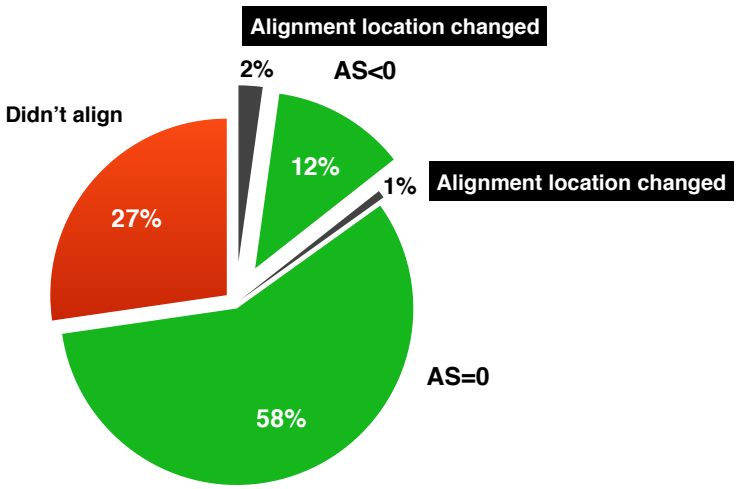


Figure 5.16 – Organization of aligned reads. Distribution of simulated reads after using prblock compressor on a collection of samples adding to 100k reads, for a compression level of 1 bit per quality score.

the calculation of the AS, such that reads that aligned once or several times, and reads that did not align, keep their alignment positions.

Accordingly, in experimenting and studying our results, it came to us to be more informative to organize reads differently. Alignment locations can be tracked implicitly in sets defined by the number of times the aligner finds an alignment location for the read. That is, changes in alignment can be explained in terms of the circulation of reads among three sets: the set of reads that aligned zero times, the set of reads that aligned one time, and the set of reads that aligned more than one time.

A portion of the following section was presented in [199]. The full content has been accepted for publication in a journal article [200].

5.9 Transparent representation of lossy quality scores: Rebinning

The hypothesis is that sequence alignment is preserved when quality score distortion and alignment score invariance occur simultaneously. To test the hypothesis, we start by reducing

Table 5.1, introduced in section 5.7, to the table shown in Figure 5.17 by grouping the quality score scale according to their penalty values for mismatches and soft-clips.

ASCII	QS	Penalties (mismatches, softclips)		Rebinning
[73]	[40]	-6	-2	**40**
[63, 72]	[30, 39]	-5	-1	**30**
[53, 62]	[20, 29]	-4	-1	**20**
[43, 52]	[10, 19]	-3	-1	**10**
[33, 42]	[0, 9]	-2	-1	**0**

Figure 5.17 – Effects of rebinning quality scores for alignment scores.

With this rebinning we can compute distortion rate baselines that represent lossy compression rates that can "at least" be applied to the quality scores of raw sequence files without compromising alignment. These baselines can be thought of as distortion thresholds, which rely on sequence files. Figure 5.18 shows the setup of our experiments. An input file with undistorted quality scores (**D**) is rebinned to produce an output file with distortion rate **d**. Both undistorted and rebinned files are aligned, and produce identical alignment reports. The distortion threshold for file **D** is **d**.

To observe the effect that quality score distortion plays on alignment we ran the three lossy compressors previously used, and set their parameters such that the output files met as close as possible the value of the distortion threshold **d**. The approximate distortion rates for each compressor are ***dA***, ***dB*** and ***dC*** (refer to Figure 5.18). The distorted files were then aligned with HISAT2 to quantify mapping results.

We experimented with synthetic and natural data and are reporting results for two natural data samples: T16M Metastatic liver tumor (whole-genome sequence data) [201], and Gene expression data in skin fibroblast cells (rna-seq data) [202]. Results are reported in the tables in Figure 5.19. The alignment report is presented as the percentage of reads grouped in one of three possible sets: reads that aligned zero times (`Z`), reads that aligned exactly one time (`X`),

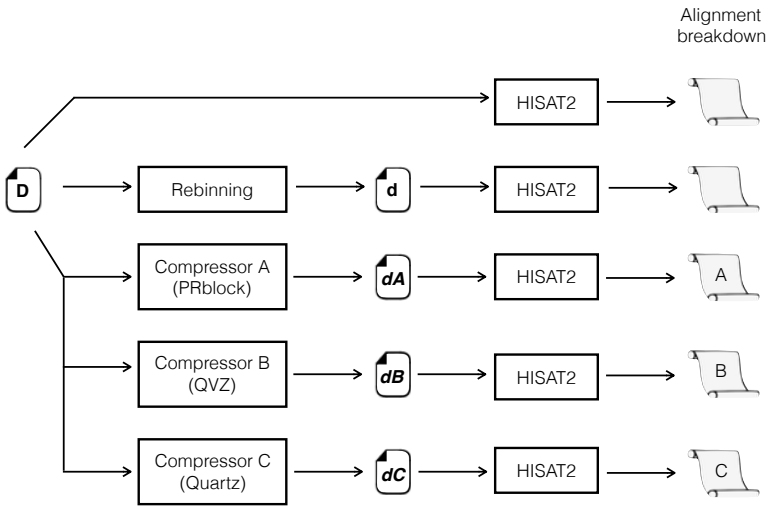


Figure 5.18 – Experimentation setup.

and reads that aligned more that one time (M).

The tables summarize alignment information as the percentage of reads whose alignment coordinate changed as a consequence of quality score distortion. We call this read relocation, and can happen between alignment sets or within alignment set M (see Figure 5.20).

For example, a read aligned before quality score distortion may be grouped in set Z but if that same reads is aligned after quality score distortion it may be grouped in set X. This type of read relocation is between sets, or interset, and the percentage of reads relocated in this fashion is shown under Interset read relocation in Figure 5.19.

The second form of read relocation can occur within set M, when the quality scores of a read with multiple alignment locations are modified in a way such that the new alignment coordinate belongs to the set of its multiple candidate locations. The percentage of reads relocated within set M is shown under Intraset read relocation in Figure 5.19. The percentages shown are relative to the total file and to the set of multireads (M).

Note that this type of read relocation occurs even in the rebinned file. This happens when the set M contains reads whose set of alignment coordinates have the same alignment score. HISAT2 will select one of the candidate coordinates for each read (primary alignment) by

rna-seq, 10k reads

Distortion method	Parameters	Distortion rate [bits/QS]	Alignment set [% reads]			Read relocation [% reads]		
			Z	X	M	Interset	Intraset	
							Total file	M
Undistorted	–	0.7715	**1.6**	**73.8**	**24.6**	–	–	
Rebinning	–	**0.3702**	**1.6**	**73.8**	**24.6**	–	0.8	3.2
PRblock	q=2, l=20	0.4028	1.7	73.9	24.4	0.4	1.1	4.5
QVZ	0.013	0.3906	1.9	73.8	24.3	0.4	0.8	3.2
Quartz	–	0.5067	1.7	77.2	21.1	3.6	2.0	9.4

wgs, 1M reads

Undistorted	–	2.696631	**6.11**	**79.21**	**14.68**	–	–	
Rebinning	–	**1.202877**	**6.11**	**79.21**	**14.68**	–	0.5037	3.4331
PRblock	q=2, l=7	1.189309	6.86	78.73	14.41	16.86	7.91	53.91
QVZ	0.035	1.202382	6.61	78.95	14.44	7.97	0.448	3.056
Quartz	–	2.465246	6.13	79.29	14.58	0.14	0.44	3.056

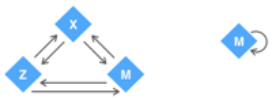


Figure 5.19 – Distortion rate and alignment percentages for wgs and rna-seq samples.

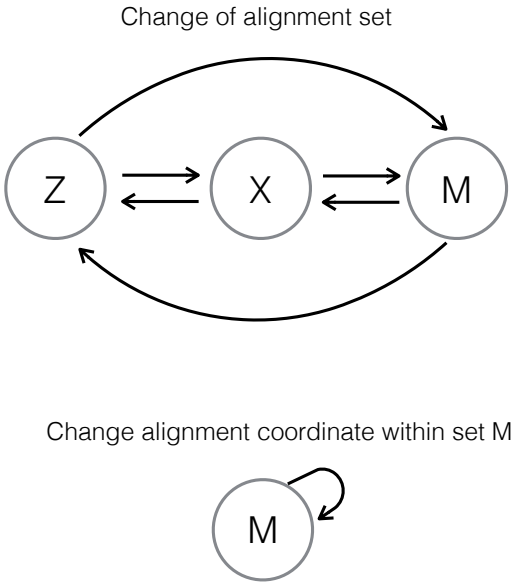


Figure 5.20 – Read relocation between sets (top), and within set `M` (bottom).

computing a pseudo-random number generated from the read name, the sequence string, the quality score string and an optional seed value. Thus, modifying the quality scores will trigger HISAT2 intrinsic response toward multireads with equally likely alignment coordinates.

The graphs in Figure 5.21 report the effect of rebinning. In both graphs, the points to the far right show the lossless compression rate for each file. In this case, no changes to the quality scores are made, and therefore, all alignment coordinates are preserved.

If we then rebin the quality score scale by changing their values according to Figure 5.17, in five bins, the file can be compressed at a rate indicated by the green dots in both graphs in Figure 5.21. The green dots identify distortion thresholds for AS invariance, and every rebinned file has a specific threshold value. Notice there is a percenage of alignment coordinate changes, which result as a consequence of intraset read relocation of multireads. In Figure 5.21, the red points to the left of the green dots show the rebin of the quality score scale but this time using three and two bins, instead of the standard five bins shown in Figure 5.17. Notice the abrupt raise in the percentage of affected reads as we move toward the left of the graph, a consequence of pushing for a coarser representation for the quality score scale.

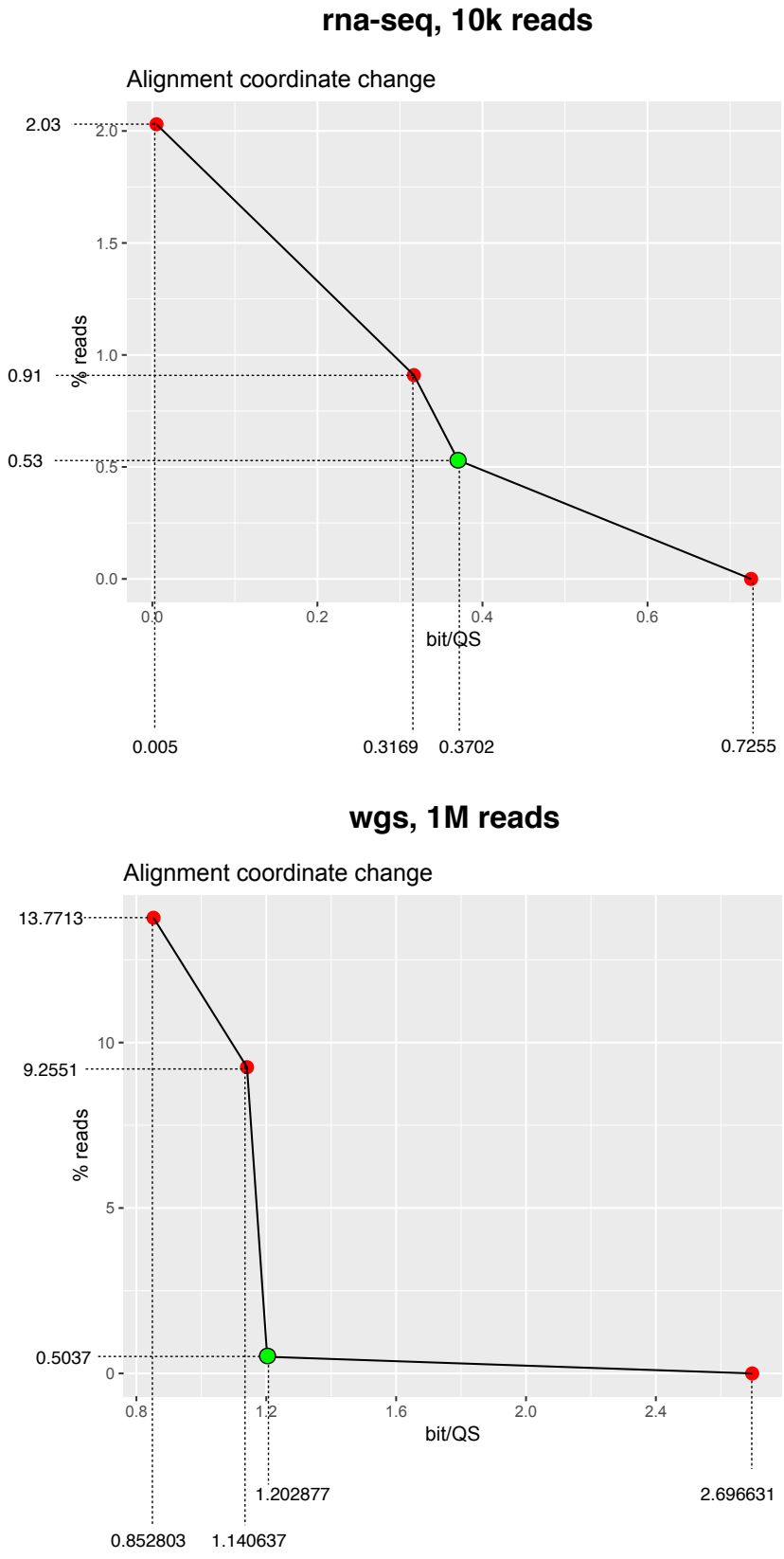


Figure 5.21 – Alignment coordinate changes for rna-seq and wgs samples

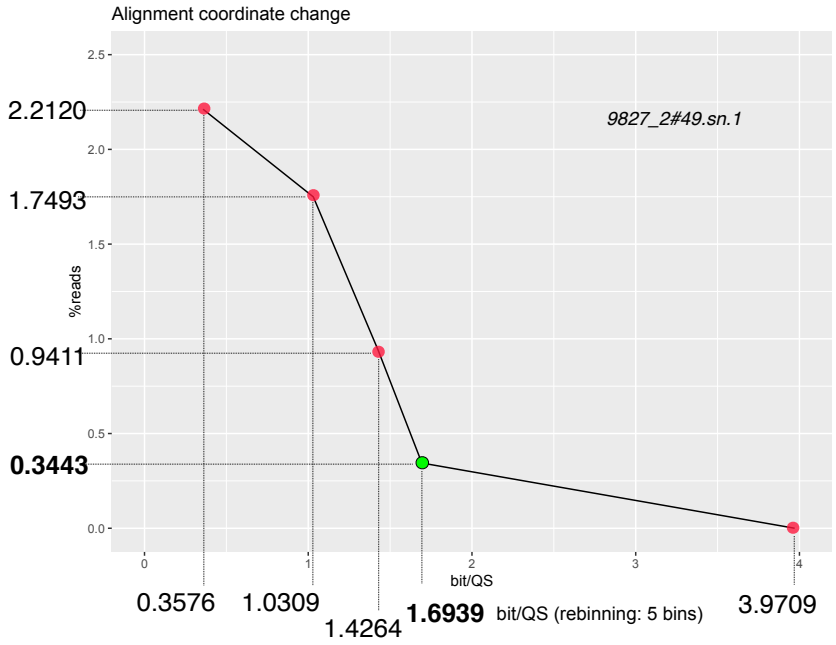


Figure 5.22 – Alignment coordinate change for sample 9827_2# 49.sn.1

In the context of [94], additional tests were run and are reported in Figures 5.22, 5.23, and 5.24.

5.10 Putting it all together

Changes to the quality scores of read sequences will inevitably lead to changes in alignment coordinates, therefore impacting alignment. Assessing the significance of this impact will depend on the recipient application following sequence alignment. However, the impact of lossy quality scores on alignment can be eliminated by keeping the alignment scores invariant. Although this is in principle true, we discovered that some idiosyncratic design decisions in the aligner weigh in unexpectedly, and collaterally impact alignment locations; this is beyond our control.

Sequence reads will fall in one of three sets after alignment, and an alignment location(s) will be reported afterward for each read. As seen at the top of Figure 5.25, each read `U` will be assigned an alignment score `AS` by the aligner, and this value will determine whether the read receives an alignment position or not. Aligned reads are grouped by the number of locations

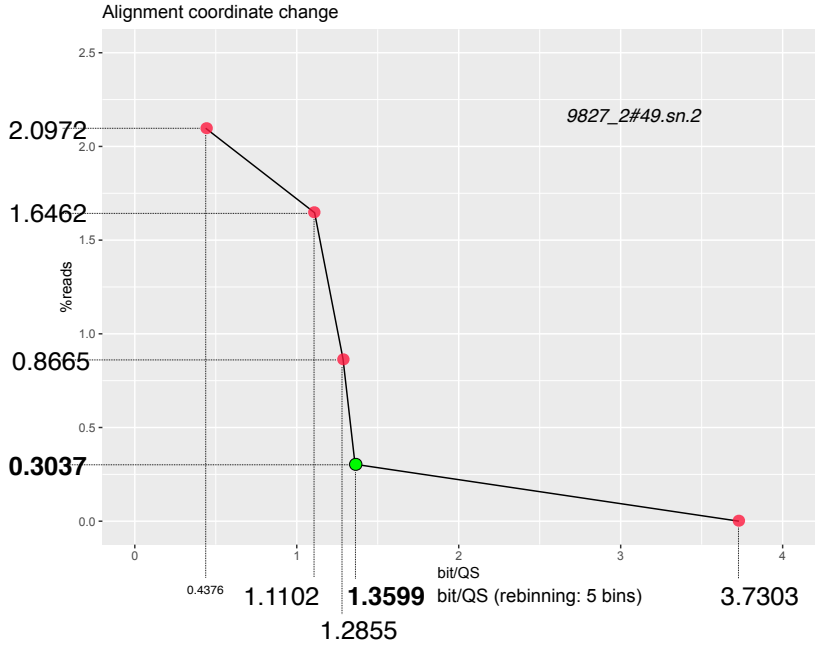


Figure 5.23 – Alignment coordinate change for sample 9827_2# 49.sn.2

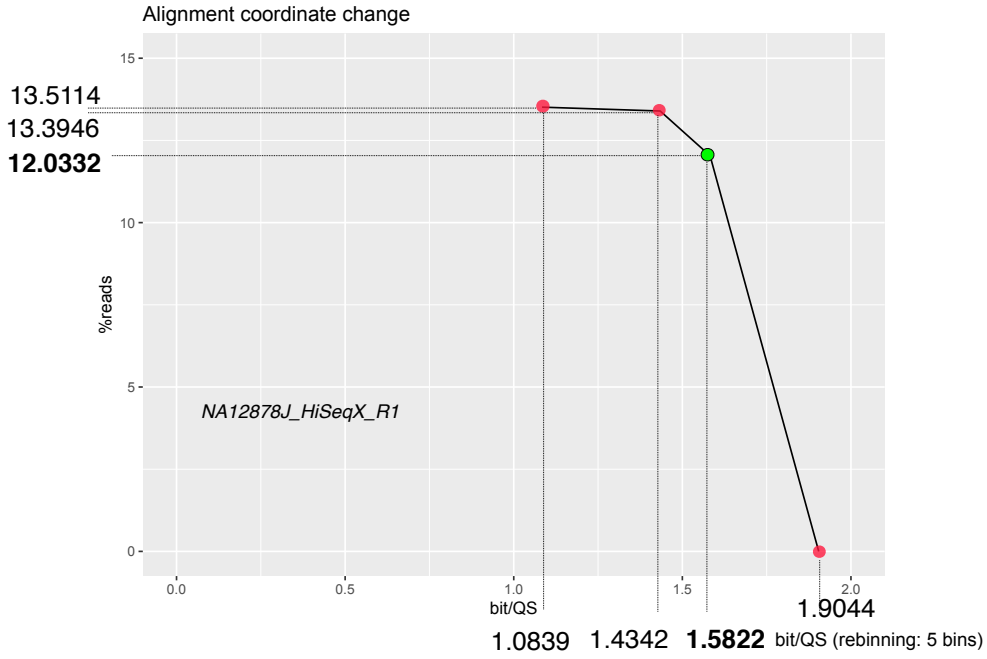


Figure 5.24 – Alignment coordinate change for sample NA12878J_HiSeqX_R1

the aligner found for them; if that number is one, they group in set `X`, if more than one location are found for a read, they group in set `M`. An unaligned read has no alignment location and belong to set `Z`.

When changes are made to their quality scores of a read `U*`, and it is then aligned, the report of its alignment score may change (`AS*`), or keep the same value (`AS`). Refer to the bottom of Figure 5.25. A change in the value of the AS does not immediately yield a change in the alignment position for that read. However, it could be the case that it does (`pos*`), and thus the alignment coordinate is tracked to record its displacement. Regardless of the outcome, the read will group in either `Z`, `X` or `M` set.

The effect of rebinning reads, in accordance to Figure 5.17, and aligning them afterward is shown in Figure 5.26. Invariance of alignment scores is achieved for every input read `U*`, and the only reads that could potentially be affected by this new representation to their quality scores are the multireads. Therefore, changes to alignment coordinates can happen only within set `M`.

A graphical summary of the process is presented in Figure 5.27. Input reads without changes, and with changes (rebinning), to their quality scores are aligned and compared side to side. The content of the three alignment sets (`Z`, `X` and `M`) are preserved. As for the alignment coordinates, they are kept unchanged with no guarantees for those in the mutiread set.

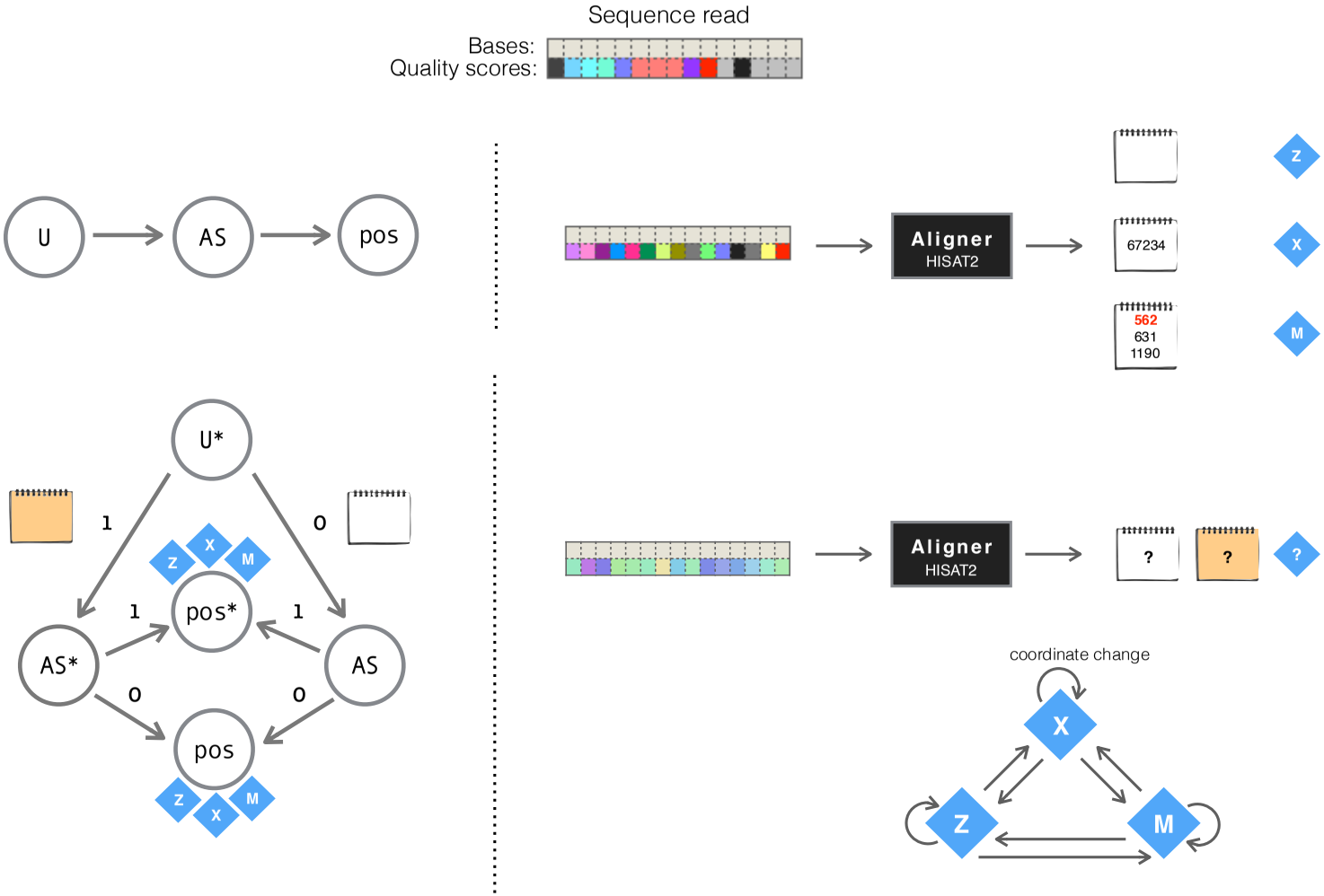


Figure 5.25 – Abstraction of the assignment of alignment location(s) for each input read. In the figure, alignment coordinates are depicted by the notepads. A read aligned as a multiread is reported with list of possible alignment locations, and one of them is randomly selected as the primary alignment (shown in red).

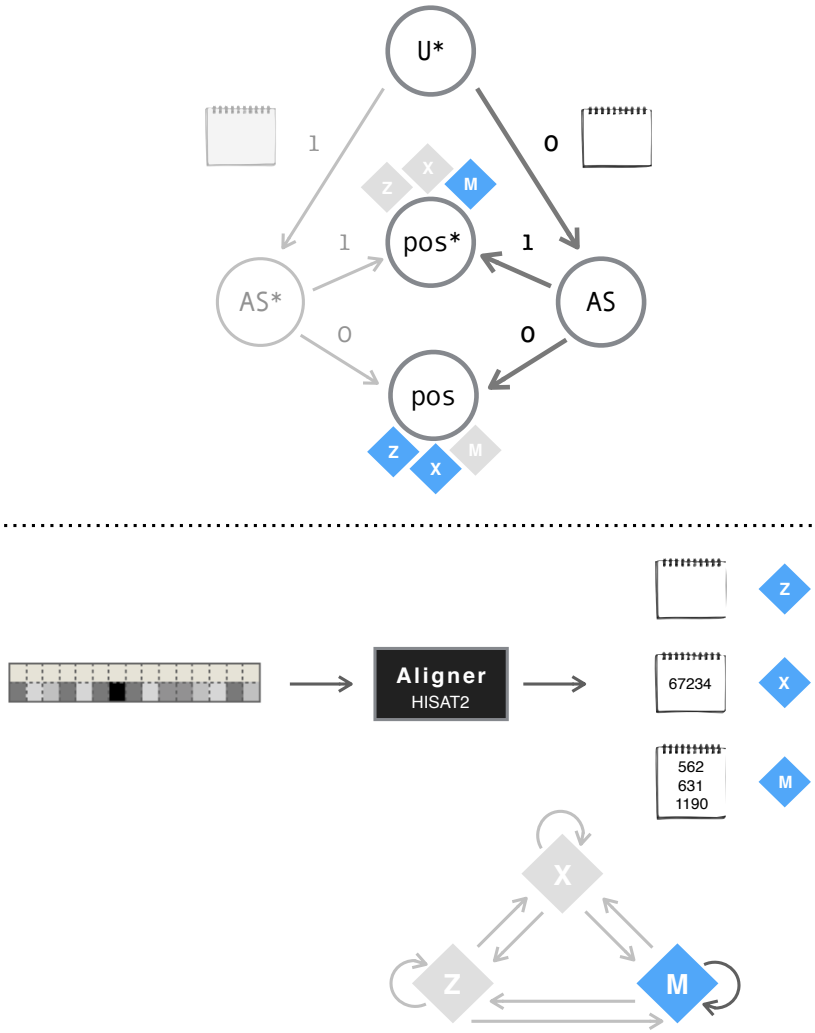


Figure 5.26 – Abstraction of the assignment of alignment location(s) for rebinned reads. For multi reads, the report of the primary alignment is randomly selected by the aligner. No guarantees can therefore be given with respect to their values.

Figure 5.27 – Schematic comparing the effect of rebinned reads on alignment.

5.11 Discussion

We investigated the penalty functions that drive the alignment score system for read sequence alignment in a well-known quality-aware aligner. We then derived a simplification in the assignment of penalty values that reduces quality score scale granularity while keeping alignment scores unaffected. Consequently, this coarser quality score scale reduces storage footprint of sequence files with the advantage of entirely preserving read mapping percentages. In other words, we distorted quality scores without collateral impact on alignment.

The aligner in question is HISAT2, the modern version of the popular aligner Bowtie2, suitable for mapping genome and exome sequence data. Compared to other quality-aware aligners like Novoalign, HISAT2's approach to alignment score computation is straight-forward and deterministic, making it a good candidate to explore the relation and effect of quality scores and sequence alignment.

Simplifying the representation of quality scores is arguably a natural choice in the face of the sequence data explosion, and computational methods that approach the problem introduce collateral errors that are difficult to quantify.

As we have discussed in previous chapters, the assessment of quality score distortion has been attempted in some application domains [42, 114, 157] without clear consensus on the limits of "safe" lossy distortion levels. Meanwhile the increasing complexity of genomic assays, datasets and computational methods only adds to the difficulty of its potential quantification.

Nevertheless, even uniform requantization of the quality scores is a suitable approximation for high accuracy applications [39], and we have shown that this approach can be extended further to rebin coarsely quality scores without impact in sequence alignment.

In the light of the fast-paced sequencing technology progress, the utility of quality scores is at stake, as they are arguably unnecessary for many omics applications. We must therefore advocate for a feasible and pertinent granularity that suits each host application.

6 Concluding remarks

We argue that today's data deluge is not the pressing problem in the biomedical sciences. With the introduction of microarray technology two decades ago, the life sciences was exposed to large amounts of data that required quantitative analysis [203]. The big data problem has only aggravated with the advent of high-throughput sequencing and the applications powered by it.

Meaningful interpretation of sequence data is becoming of crucial importance. As the democratization of high-throughput sequencing analysis carries on, the real challenge is to carry out computational analyses on the vast amounts of data available. Data interpretation must become as accessible as data generation to sustain the growing number of applications powered by sequence data. To this end, practical storage solutions need not only continue to be developed but also need to be disseminated and successfully adopted.

The constant flux and expanding scope of high-throughput sequencing analysis has complicated the development of best practices that could facilitate the use of heterogeneous software. There is a lack of general agreement on how analyses are to be carried out [47, 184, 103, 175]. Moreover, existing best practices may be too elaborate for many researches who opt for more straightforward approaches, or decide instead to use alternative computational methods that may yield comparable results [175]. How to approach this scenario of computationally

complex analyses, lax standards of computational guidelines, and huge storage footprints of sequence data?

We believe that enough research work has been developed in the last years to support the case for lossy representation for the quality scores as the key to substantially reduce storage footprints in sequence data. Simplifying the representation of quality scores is a natural choice in the face of the sequence data explosion. Nevertheless, the computational methods that approach the problem introduce collateral errors that are difficult to quantify.

The assessment of quality score distortion in genomic sequence data has been attempted for DNA (variant calling) and RNA (differential gene expression) without clear consensus on the limits of “safe” lossy distortion levels. Meanwhile, the increasing complexity of genomic assays, datasets and computational methods only adds to the difficulty of potentially quantifying “safe” quality score distortion levels.

For read alignment of DNA and RNA sequence data, it is possible to compute a threshold value for transparent quality score distortion, which allows the identification of a "safe" representation for quality score values. To achieve this, the quality score scale is rebinned in compliance to the penalty functions governing alignment scores. The threshold, expressed in bits per quality score, is identified for alignment score invariance in the aligner HISAT2, and its value is distinct for each rebinne file.

Originally we stood by the assumptions that the overhead in changing quality score representation, along with the collateral effect caused to omics analyses may be the practical limiting factors for the adoption of lossy schemes. However, after following the progression in the field and familiarizing with the ‘unwritten rules’ in sequence data analysis and processing we have a newfound impression.

We presented detailed evidence of marginal effects in the application of lossy quality scores in omics analyses, a result that supports and corroborates what is reported in the literature. Therefore, we reconcile the idea that the impact produced by lossy quality score representation prevent the practical application of lossy schemes in omics analyses. What is limiting their use

is, instead, the lack of awareness of lossy approaches inside biomedical research circles, where compression of sequence data is seemingly assumed lossless. Consequently, lossy quality score compression is hardly mentioned [81], or it is vaguely referred to [31], or it is straight-out not considered [30, 47].

The future in omics analysis will be the widespread adoption of integrative frameworks that amalgamate application-specific tools to achieve more accurate results. This should allow the use and implementation of best practices and standards to represent, manipulate and process sequence data, and track their details.

As for the quality scores, we posit that they will be imminently discarded. A coarser scale for their values were rolled out two years ago by the primary manufacturer of sequencing platforms [52]. This new sequence data uses four values for the full quality score scale, a reduction of 50% to the previous optional 8-level binning [204]. Future research directions for compression of sequence data should be in emergent problems in the integration and manipulation of multiomics datasets [181].

Bibliography

[1] "The GovLab Index: The Data Universe." [ONLINE] Available at: http://thegovlab.org/govlab-index-the-digital-universe/. [Accessed June 18, 2019].

[2] "A day in Big Data - For Smarter Customer Experiences - OgilvyOne." [ONLINE] Available at: http://adayinbigdata.com/. [Accessed June 18, 2019].

[3] "The Flood of Big Data." [ONLINE] Available at: https://www.ibmbigdatahub.com/infographic/flood-big-data. [Accessed June 18, 2019].

[4] "The Digital Universe Rich Data and the Increasing Value of the Internet of Things." [ONLINE] Available at: https://www.emc.com/collateral/analyst-reports/idc-digital-universe-2014.pdf. [Accessed June 18, 2019].

[5] M. Tanwar, R. Duggal, and S. K. Khatri, "Unravelling unstructured data: A wealth of information in big data," in *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, Sept. 2015.

[6] R. Kitchin, "Big Data, new epistemologies and paradigm shifts," *Big Data & Society*, no. 1, 2014.

[7] K. Taylor-Sakyi, "Big Data: Understanding Big Data," *arXiv preprint arXiv:1601.04602*, 2016.

[8] "Government Office for Science. The Internet of Things: making the most of the Second Digital Revolution.." [ONLINE] Available at: https://www.gov.uk/government/uploads/

system/uploads/attachment_data/file/409774/14-1230-internet-of-things-review.pdf. [Accessed June 18, 2019].

[9] "IDC Forecasts Worldwide Spending on the Internet of Things." [ONLINE] Available at: https://www.idc.com/getdoc.jsp?containerId=prUS44596319. [Accessed June 20, 2019].

[10] A. Holzinger, C. Stocker, B. Ofner, G. Prohaska, A. Brabenetz, and R. Hofmann-Wellenhof, "Combining HCI, Natural Language Processing, and Knowledge Discovery - Potential of IBM Content Analytics as an Assistive Technology in the Biomedical Field," in *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013.

[11] N. Khan, I. Yaqoob, I. A. T. Hashem, Z. Inayat, W. K. Mahmoud Ali, M. Alam, M. Shiraz, and A. Gani, "Big Data: Survey, Technologies, Opportunities, and Challenges," *The Scientific World Journal*, 2014.

[12] C. Hammer, D. Kostroch, and G. Quiros, "Big Data: Potential, Challenges and Statistical Implications," *Staff Discussion Notes*, 2017.

[13] Z. D. Stephens, S. Y. Lee, F. Faghri, R. H. Campbell, C. Zhai, M. J. Efron, R. Iyer, M. C. Schatz, S. Sinha, and G. E. Robinson, "Big Data: Astronomical or Genomical?," *PLOS Biology*, July 2015.

[14] M. Eisenstein, "Big data: The power of petabytes," *Nature*, no. 7576, 2015.

[15] V. Costa, C. Angelini, I. De Feis, and A. Ciccodicola, "Uncovering the Complexity of Transcriptomes with RNA-Seq," *Journal of Biomedicine and Biotechnology*, 2010.

[16] A. Kuo, "Mining Health Big Data - Opportunities and Challenges." [ONLINE] Available at: https://bigdata.ieee.org/images/files/pdf/Health2.0-China---Mining-healthcare-Big-Data.pptx.pdf. [Accessed June 22, 2019].

[17] IHTT, "Transforming Health Care through Big Data Strategies for leveraging big data in the health care industry," tech. rep., 2013.

[18] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: promise and potential," *Health Information Science and Systems*, 2014.

[19] J. Sun and C. Reddy, "Big Data Analytics for Healthcare. Tutorial presentation at SIGKDD." [ONLINE] Available at: http://dmkd.cs.vt.edu/TUTORIAL/Healthcare/part1.pdf. [Accessed June 22, 2019].

[20] H. Chang, "Book Review: Data-Driven Healthcare & Analytics in a Big Data World," *Healthcare Informatics Research*, 2015.

[21] K. Wetterstrand, "DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP)." [ONLINE] Available at: https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data. [Accessed July 2, 2019].

[22] E. Birney, J. Vamathevan, and P. Goodhand, "Genomics in healthcare: GA4GH looks to 2022," *bioRxiv*, 2017.

[23] "Genomics England. the 100,000 Genomes Project." [ONLINE] Available at: https://www.genomicsengland.co.uk/about-genomics-england/the-100000-genomes-project/, 2014. [Accessed June 22, 2019].

[24] "Next in the Genomics Revolution: The Era of the Social Genome | Veritas Genetics." [ONLINE] Available at: https://www.veritasgenetics.com/next-genomics-revolution-era-social-genome. [Accessed June 22, 2019].

[25] "DNA Sequencing Methods Collection. An overview of recent DNA-seq publications featuring Illumina technology." [ONLINE] Available at: https://www.illumina.com/content/dam/illumina-marketing/documents/products/research_reviews/dna-sequencing-methods-review-web.pdf. [Accessed June 22, 2019].

[26] J. Reuter, D. V. Spacek, and M. Snyder, "High-Throughput Sequencing Technologies," *Molecular Cell*, no. 4, 2015.

[27] K. He, D. Ge, and M. He, "Big Data Analytics for Genomic Medicine," *International Journal of Molecular Sciences*, 2017.

[28] J. Davis-Turak, S. M. Courtney, E. S. Hazard, W. B. Glen, W. A. da Silveira, T. Wesselman, L. P. Harbin, B. J. Wolf, D. Chung, and G. Hardiman, "Genomics pipelines and data integration: challenges and opportunities in the research setting," *Expert Review of Molecular Diagnostics*, 2017.

[29] B. Berger, J. Peng, and M. Singh, "Computational solutions for omics data," *Nature Reviews Genetics*, 2013.

[30] L. Papageorgiou, P. Eleni, S. Raftopoulou, M. Mantaiou, V. Megalooikonomou, and D. Vlachakis, "Genomic big data hitting the storage bottleneck," *EMBnet.journal*, 2018.

[31] M. Hosseini, D. Pratas, and A. Pinho, "A Survey on Data Compression Methods for Biological Sequences," *Information*, 2016.

[32] S. Levy, G. Sutton, P. C. Ng, L. Feuk, A. L. Halpern, B. P. Walenz, N. Axelrod, J. Huang, E. F. Kirkness, G. Denisov, Y. Lin, J. R. MacDonald, A. W. C. Pang, M. Shago, T. B. Stockwell, A. Tsiamouri, V. Bafna, V. Bansal, S. A. Kravitz, D. A. Busam, K. Y. Beeson, T. C. McIntosh, K. A. Remington, J. F. Abril, J. Gill, J. Borman, Y.-H. Rogers, M. E. Frazier, S. W. Scherer, R. L. Strausberg, and J. C. Venter, "The Diploid Genome Sequence of an Individual Human," *PLoS Biology*, 2007.

[33] P.-R. Loh, M. Baym, and B. Berger, "Compressive genomics," *Nature Biotechnology*, 2012.

[34] I. Numanagić, J. K. Bonfield, F. Hach, J. Voges, J. Ostermann, C. Alberti, M. Mattavelli, and S. C. Sahinalp, "Comparison of high-throughput sequencing data compression tools," *Nature Methods*, 2016.

[35] M. Hsi-Yang Fritz, R. Leinonen, G. Cochrane, and E. Birney, "Efficient storage of high throughput DNA sequencing data using reference-based compression," *Genome Research*, 2011.

[36] C. Alberti, T. Paridaens, J. Voges, D. Naro, J. J. Ahmad, M. Ravasi, D. Renzi, G. Zoia, I. Ochoa, M. Mattavelli, J. Delgado, and M. Hernaez, "An introduction to MPEG-G, the new ISO standard for genomic information representation," *bioRxiv*, 2018.

[37] C. Kozanitis, C. Saunders, S. Kruglyak, V. Bafna, and G. Varghese, "Compressing Genomic Sequence Fragments Using SlimGene," *Journal of Computational Biology*, 2011.

[38] J. Shendure, S. Balasubramanian, G. M. Church, W. Gilbert, J. Rogers, J. A. Schloss, and R. H. Waterston, "DNA sequencing at 40: past, present and future," *Nature*, 2019.

[39] "Illumina white paper. Reducing Whole-Genome Data Storage Footprint.," tech. rep., 2012.

[40] D. C. Jones, W. L. Ruzzo, X. Peng, and M. G. Katze, "Compression of next-generation sequencing reads aided by highly efficient de novo assembly," *Nucleic Acids Research*, 2012.

[41] F. Hach, I. Numanagić, C. Alkan, and S. C. Sahinalp, "SCALCE: boosting sequence compression algorithms using locally consistent encoding," *Bioinformatics*, 2012.

[42] I. Ochoa, M. Hernaez, R. Goldfeder, T. Weissman, and E. Ashley, "Effect of lossy compression of quality scores on variant calling," *Briefings in Bioinformatics*, 2016.

[43] C. Xu, "A review of somatic single nucleotide variant calling algorithms for next-generation sequencing data," *Computational and Structural Biotechnology Journal*, 2018.

[44] I. Y. Abdurakhmonov, "Bioinformatics: Basics, Development, and Future," *IntechOpen*, 2016.

[45] D. L. Greenfield, O. Stegle, and A. Rrustemi, "GeneCodeq: quality score compression and improved genotyping using a Bayesian framework," *Bioinformatics*, 2016.

[46] Å. Roguski, I. Ochoa, M. Hernaez, and S. Deorowicz, "FaStore: a space-saving solution for raw sequencing data," *Bioinformatics*, 2018.

[47] A. A. Regier, Y. Farjoun, D. E. Larson, O. Krasheninina, H. M. Kang, D. P. Howrigan, B.-J. Chen, M. Kher, E. Banks, D. C. Ames, A. C. English, H. Li, J. Xing, Y. Zhang, T. Matise, G. R. Abecasis, W. Salerno, M. C. Zody, B. M. Neale, and I. M. Hall, "Functional equivalence

of genome sequencing analysis pipelines enables harmonized variant calling across human genetics projects," *Nature Communications*, 2018.

[48] S. Mangul, L. S. Martin, B. L. Hill, A. K.-M. Lam, M. G. Distler, A. Zelikovsky, E. Eskin, and J. Flint, "Systematic benchmarking of omics computational tools," *Nature Communications*, 2019.

[49] "Illumina. RNA Sequencing Methods Collection, An overview of recent RNA-Seq publications featuring Illumina technology," tech. rep., 2017.

[50] V. J. Henry, A. E. Bandrowski, A.-S. Pepin, B. J. Gonzalez, and A. Desfeux, "OMICtools: an informative directory for multi-omic data analysis," *Database*, 2014.

[51] S. Chandak, K. Tatwawadi, I. Ochoa, M. Hernaez, and T. Weissman, "SPRING: a next-generation compressor for FASTQ data," *Bioinformatics (Oxford, England)*, vol. 35, pp. 2674–2676, Aug. 2019.

[52] "Illumina. NovaSeq 6000 System Quality Scores and RTA3 Software," tech. rep., 2017.

[53] S. Sandmann, A. O. de Graaf, M. Karimi, B. A. van der Reijden, E. HellstrÃ¶m-Lindberg, J. H. Jansen, and M. Dugas, "Evaluating Variant Calling Tools for Non-Matched Next-Generation Sequencing Data," *Scientific Reports*, 2017.

[54] X. Yu, K. Guda, J. Willis, M. Veigl, Z. Wang, S. Markowitz, M. D. Adams, and S. Sun, "How do alignment programs perform on sequencing data with varying qualities and from repetitive regions?," *BioData Mining*, 2012.

[55] C. Firtina and C. Alkan, "On genomic repeats and reproducibility," *Bioinformatics*, 2016.

[56] A. Cornish and C. Guda, "A Comparison of Variant Calling Pipelines Using Genome in a Bottle as a Reference," *BioMed Research International*, 2015.

[57] J. Voges, A. Fotouhi, J. Ostermann, and K. M. Oğuzhan, "A Two-Level Scheme for Quality Score Compression," *Journal of Computational Biology*, 2018.

[58] "Illumina. DNA Sequencing Methods Collection, An overview of recent DNA-Seq publications featuring Illumina technology," tech. rep., 2017.

[59] K.-B. Hwang, I.-H. Lee, H. Li, D.-G. Won, C. Hernandez-Ferrer, J. A. Negron, and S. W. Kong, "Comparative analysis of whole-genome sequencing pipelines to minimize false negative findings," *Scientific Reports*, 2019.

[60] J. K. Kulski, "Next-Generation Sequencing — An Overview of the History, Tools, and "Omic" Applications," *Next Generation Sequencing - Advances, Applications and Challenges*, 2016.

[61] F. Sanger, S. Nicklen, and A. R. Coulson, "DNA sequencing with chain-terminating inhibitors," *Proceedings of the National Academy of Sciences*, 1977.

[62] A. M. Maxam and W. Gilbert, "A new method for sequencing DNA.," *Proceedings of the National Academy of Sciences*, 1977.

[63] M. Kchouk, J. F. Gibrat, and M. Elloumi, "Generations of Sequencing Technologies: From First to Next Generation," *Biology and Medicine*, 2017.

[64] S. Pillai, V. Gopalan, and A. K.-Y. Lam, "Review of sequencing platforms and their applications in phaeochromocytoma and paragangliomas," *Critical Reviews in Oncology/Hematology*, 2017.

[65] S. Goodwin, J. D. McPherson, and W. R. McCombie, "Coming of age: ten years of next-generation sequencing technologies," *Nature Reviews. Genetics*, 2016.

[66] Y. O. Alekseyev, R. Fazeli, S. Yang, R. Basran, T. Maher, N. S. Miller, and D. Remick, "A Next-Generation Sequencing Primer—How Does It Work and What Can It Do?," *Academic Pathology*, 2018.

[67] D. A. Wheeler, M. Srinivasan, M. Egholm, Y. Shen, L. Chen, A. McGuire, W. He, Y.-J. Chen, V. Makhijani, G. T. Roth, X. Gomes, K. Tartaro, F. Niazi, C. L. Turcotte, G. P. Irzyk, J. R. Lupski, C. Chinault, X.-z. Song, Y. Liu, Y. Yuan, L. Nazareth, X. Qin, D. M. Muzny,

M. Margulies, G. M. Weinstock, R. A. Gibbs, and J. M. Rothberg, "The complete genome of an individual by massively parallel DNA sequencing," *Nature*, 2008.

[68] J. M. Heather and B. Chain, "The sequence of sequencers: The history of sequencing DNA," *Genomics*, 2016.

[69] C. S. Pareek, R. Smoczynski, and A. Tretyn, "Sequencing technologies and genome sequencing," *Journal of Applied Genetics*, 2011.

[70] J. Shendure, S. Balasubramanian, G. M. Church, W. Gilbert, J. Rogers, J. A. Schloss, and R. H. Waterston, "DNA sequencing at 40: past, present and future," *Nature*, 2017.

[71] M. Di Ventra and M. Taniguchi, "Decoding DNA, RNA and peptides with quantum tunnelling," *Nature Nanotechnology*, 2016.

[72] J. Wilson, L. Sloman, Z. He, and A. Aksimentiev, "Graphene Nanopores for Protein Sequencing," *Advanced Functional Materials*, 2016.

[73] P. L. Ståhl, F. Salm'en, S. Vickovic, A. Lundmark, J. F. Navarro, J. Magnusson, S. Giacomello, M. Asp, J. O. Westholm, M. Huss, A. Mollbrink, S. Linnarsson, S. Codeluppi, Å. Borg, F. Pontén, P. I. Costea, P. Sahlén, J. Mulder, O. Bergmann, J. Lundeberg, and J. Frisén, "Visualization and analysis of gene expression in tissue sections by spatial transcriptomics," *Science*, 2016.

[74] E. D. Green and M. S. Guyer, "Charting a course for genomic medicine from base pairs to bedside," *Nature*, 2011.

[75] P. J. A. Cock, C. J. Fields, N. Goto, M. L. Heuer, and P. M. Rice, "The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants," *Nucleic Acids Research*, 2010.

[76] B. Ewing, L. Hillier, M. C. Wendl, and P. Green, "Base-Calling of Automated Sequencer Traces Using *Phred*. I. Accuracy Assessment," *Genome Research*, 1998.

[77] I. Inc., “Quality scores for next-generation sequencing. Assesing sequencing accuracy using Phred quality scoring,” technical Note, 2011.

[78] S. Deorowicz and S. Grabowski, “Data compression for sequencing data,” *Algorithms for Molecular Biology : AMB*, 2013.

[79] “MPEG Requirements, "ISO/IEC JTC1/SC29/WG11 MPEG2015/N15739 - Evaluation framework of lossy compression of Quality Values".” [ONLINE] Available at: http://mpeg.chiariglione.org/standards/exploration/genome-compression.

[80] I. Ochoa, M. Hernaez, and T. Weissman, “Aligned genomic data compression via improved modeling,” *Journal of Bioinformatics and Computational Biology*, 2014.

[81] A. El Allali, “MZPAQ: a FASTQ data compression tool,” *Source Code for Biology and Medicine*, 2019.

[82] J. K. Bonfield and M. V. Mahoney, “Compression of FASTQ and SAM Format Sequencing Data,” *PLoS ONE*, 2013.

[83] “Samtools, WGS/WES Mapping to Variant Calls, htslib.org.” [ONLINE] Available at: http://www.htslib.org/workflow/.

[84] J. K. Bonfield, “The Scramble conversion tool,” *Bioinformatics*, 2014.

[85] G. Malysa, M. Hernaez, I. Ochoa, M. Rao, K. Ganesan, and T. Weissman, “QVZ: lossy compression of quality values,” *Bioinformatics*, Oct. 2015.

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

[87] Y. W. Yu, D. Yorukoglu, J. Peng, and B. Berger, “Quality score compression improves genotyping accuracy,” *Nature Biotechnology*, 2015.

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

[89] S. D. Kahn, "On the Future of Genomic Data," *Science*, vol. 331, Feb. 2011.

[90] H. Li, "SAM/BAM related specifications." [ONLINE] Available at: http://samtools.github.io/hts-specs/.

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

[92] W. R. Pearson and D. J. Lipman, "Improved tools for biological sequence comparison.," *Proceedings of the National Academy of Sciences*, 1988.

[93] V. Buffalo, *Bioinformatics data skills*. O'Reilly, first edition ed., 2015.

[94] C. Alberti, "Investigation on Genomic Information Compression and Storage ISO/IEC JTC 1/SC 29/WG 11 N15346." [ONLINE] Available at: https://mpeg.chiariglione.org/sites/default/files/files/standards/parts/docs/w15346_Investigation_on_genomic_information_compression.pdf, 2015.

[95] M. Adler, "pigz - Parallel gzip." [ONLINE] Available at: http://zlib.net/pigz/.

[96] J. Gilchrist, "Parallel data compression with bzip2," *Proceedings of the 16th IASTED international conference on parallel and distributed computing and systems*, 2004.

[97] S. Deorowicz and S. Grabowski, "Compression of DNA sequence reads in FASTQ format," *Bioinformatics*, 2011.

[98] J. Leipzig, "A review of bioinformatic pipeline frameworks," *Briefings in Bioinformatics*, 2016.

[99] E. Afgan, D. Baker, M. van den Beek, D. Blankenberg, D. Bouvier, M. Cech, J. Chilton, D. Clements, N. Coraor, C. Eberhard, B. GrÃŒning, A. Guerler, J. Hillman-Jackson, G. VonÂ Kuster, E. Rasche, N. Soranzo, N. Turaga, J. Taylor, A. Nekrutenko, and J. Goecks, "The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update," *Nucleic Acids Research*, 2016.

[100] "Illumina. An Introduction to Next-Generation Sequencing Technology," tech. rep., 2016.

[101] D. Rigden and X. Fernández, "The 26th annual Nucleic Acids Research database issue and Molecular Biology Database Collection," *Nucleic Acids Research*, 2012.

[102] D. A. Benson, M. Cavanaugh, K. Clark, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and E. W. Sayers, "GenBank," *Nucleic Acids Research*, 2012.

[103] O. Spjuth, E. Bongcam-Rudloff, G. C. Hernández, L. Forer, M. Giovacchini, R. V. Guimera, A. Kallio, E. Korpelainen, M. M. Korpelainen, M. Krachunov, D. P. Kreil, O. Kulev, P. P. Łabaj, S. Lampa, L. Pireddu, S. Schönherr, A. Siretskiy, and D. Vassilev, "Experiences with workflows for automating data-intensive bioinformatics," *Biology Direct*, 2015.

[104] A. Altmann, P. Weber, D. Bader, M. Preuss, E. B. Binder, and B. Müller-Myhsok, "A beginners guide to SNP calling from high-throughput DNA-sequencing data," *Human Genetics*, 2012.

[105] T. J. Treangen and S. L. Salzberg, "Repetitive DNA and next-generation sequencing: computational challenges and solutions," *Nature Reviews Genetics*, 2012.

[106] R. Nielsen, J. S. Paul, A. Albrechtsen, and Y. S. Song, "Genotype and SNP calling from next-generation sequencing data," *Nature Reviews Genetics*, 2011.

[107] "Broad Institute. GATK talks: Indel-based realignment.." [ONLINE] Available at: https://docs.google.com/file/d/0B2dK2q40HDWeLTFzNndsNDBuVms/preview. [Accessed July 12, 2019].

[108] "Broad Institute. GATK talks: Base quality score recalibration." [ONLINE] Available at: https://docs.google.com/file/d/0B2dK2q40HDWeZk1rMXpTYmZzTXc/preview. [Accessed July 12, 2019].

[109] Q. Liu, Y. Guo, J. Li, J. Long, B. Zhang, and Y. Shyr, "Steps to ensure accuracy in genotype and SNP calling from Illumina sequencing data," *BMC genomics*, 2012.

[110] J. O'Rawe, T. Jiang, G. Sun, Y. Wu, W. Wang, J. Hu, P. Bodily, L. Tian, H. Hakonarson, W. E. Johnson, Z. Wei, K. Wang, and G. J. Lyon, "Low concordance of multiple variant-calling pipelines: practical implications for exome and genome sequencing," *Genome Medicine*, 2013.

[111] S. Hwang, E. Kim, I. Lee, and E. M. Marcotte, "Systematic comparison of variant calling pipelines using gold standard personal exome variants," *Scientific Reports*, 2015.

[112] H. Li, "Toward better understanding of artifacts in variant calling from high-coverage samples," *Bioinformatics*, 2014.

[113] "Broad Institute. GATK talks: Annotating and Analyzing varian calls.." [ONLINE] Available at: https://docs.google.com/file/d/0B2dK2q40HDWeWi1YMm42bWdpRE0/preview. [Accessed July 14, 2019].

[114] C. Alberti, N. Daniels, M. Hernaez, J. Voges, R. L. Goldfeder, A. A. Hernandez-Lopez, M. Mattavelli, and B. Berger, "An Evaluation Framework for Lossy Compression of Genome Sequencing Quality Values," in *2016 Data Compression Conference (DCC)*, (Snowbird, UT, USA), IEEE, 2016.

[115] "Coriell Institute, "NA12878," International HapMap Project." [ONLINE] Available at: https://www.coriell.org/0/Sections/Search/Sample_Detail.aspx?Ref=GM12878.

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

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

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

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

[120] “DNA Data Bank of Japan, "SRX514833," DNA Data Bank of Japan.” [ONLINE] Available at: https://trace.ddbj.nig.ac.jp/DRASearch/experiment?acc=SRX514833.

[121] H. Li and R. Durbin, “Fast and accurate short read alignment with Burrows-Wheeler transform,” *Bioinformatics*, 2009.

[122] “Broad Institute. GATK Best Practices..” [ONLINE] Available at: https://www.broadinstitute.org/gatk/guide/best-practices. [Accessed July 14, 2019].

[123] B. Langmead and S. L. Salzberg, “Fast gapped-read alignment with Bowtie 2,” *Nature Methods*, 2012.

[124] N. D. Olson, S. P. Lund, R. E. Colman, J. T. Foster, J. W. Sahl, J. M. Schupp, P. Keim, J. B. Morrow, M. L. Salit, and J. M. Zook, “Best practices for evaluating single nucleotide variant calling methods for microbial genomics,” *Frontiers in Genetics*, 2015.

[125] B. Alberts, ed., *Molecular biology of the cell.* New York: Garland Science, 5th ed ed., 2008.

[126] N. Altman, “Measuring gene expression. Presentation at Penn State University.” [ONLINE] Available at: https://slideplayer.com/slide/6081581/. [Accessed July 22, 2019].

[127] D. Wishart, “Measuring gene expression. Presentation at the University of Alberta.” [ONLINE] Available at: https://www.gene-quantification.de/wishart-gene-expression-1.pdf. [Accessed July 22, 2019].

[128] A. Gitter, “Measuring transcriptomes with RNA-Seq . Presentation at University of Wisconsin.” [ONLINE] Available at: https://www.biostat.wisc.edu/bmi776/lectures/rnaseq.pdf. [Accessed July 22, 2019].

[129] Z. Wang, M. Gerstein, and M. Snyder, “RNA-Seq: a revolutionary tool for transcriptomics,” *Nature Reviews Genetics*, 2009.

[130] S. M. E. Sahraeian, M. Mohiyuddin, R. Sebra, H. Tilgner, P. T. Afshar, K. F. Au, N. Bani Asadi, M. B. Gerstein, W. H. Wong, M. P. Snyder, E. Schadt, and H. Y. K. Lam,

"Gaining comprehensive biological insight into the transcriptome by performing a broad-spectrum RNA-seq analysis," *Nature Communications*, 2017.

[131] Y. Han, S. Gao, K. Muegge, W. Zhang, and B. Zhou, "Advanced Applications of RNA Sequencing and Challenges," *Bioinformatics and Biology Insights*, 2015.

[132] M. Garber, M. G. Grabherr, M. Guttman, and C. Trapnell, "Computational methods for transcriptome annotation and quantification using RNA-seq," *Nature Methods*, 2011.

[133] E. L. van Dijk, Y. Jaszczyszyn, and C. Thermes, "Library preparation methods for next-generation sequencing: Tone down the bias," *Experimental Cell Research*, 2014.

[134] M. Griffith, J. R. Walker, N. C. Spies, B. J. Ainscough, and O. L. Griffith, "Informatics for RNA Sequencing: A Web Resource for Analysis on the Cloud," *PLOS Computational Biology*, 2015.

[135] K. R. Kukurba and S. B. Montgomery, "RNA Sequencing and Analysis," *Cold Spring Harbor Protocols*, 2015.

[136] F. Dündar, "Introduction to differential gene expression analysis using RNA-seq," 2015.

[137] G. Litwack, *Human biochemistry*. Amsterdam ; Boston: Academic Press, 2018.

[138] P. G. Engström, T. Steijger, B. Sipos, G. R. Grant, A. Kahles, G. Rätsch, N. Goldman, T. J. Hubbard, J. Harrow, R. Guigó, and P. Bertone, "Systematic evaluation of spliced alignment programs for RNA-seq data," *Nature Methods*, 2013.

[139] D. Sims, I. Sudbery, N. E. Ilott, A. Heger, and C. P. Ponting, "Sequencing depth and coverage: key considerations in genomic analyses," *Nature Reviews Genetics*, 2014.

[140] Y. Liu, J. Zhou, and K. P. White, "RNA-seq differential expression studies: more sequence or more replication?," *Bioinformatics*, 2014.

[141] K. D. Hansen, Z. Wu, R. A. Irizarry, and J. T. Leek, "Sequencing technology does not eliminate biological variability," *Nature Biotechnology*, 2011.

[142] "The ENCODE Consortium. Standards, Guidelines and Best Practices for RNA-Seq." [ONLINE] Available at: https://www.encodeproject.org/documents/cede0cbe-d324-4ce7-ace4-f0c3eddf5972/@@download/attachment/ENCODE%20Best%20Practices%20for%20RNA_v2.pdf.

[143] L. Pachter, "Models for transcript quantification from RNA-Seq," *arXiv:1104.3889*, 2011.

[144] A. Oshlack, M. D. Robinson, and M. D. Young, "From RNA-seq reads to differential expression results," *Genome Biology*, 2010.

[145] G. Baruzzo, K. E. Hayer, E. J. Kim, B. Di Camillo, G. A. FitzGerald, and G. R. Grant, "Simulation-based comprehensive benchmarking of RNA-seq aligners," *Nature Methods*, 2017.

[146] N. L. Bray, H. Pimentel, P. Melsted, and L. Pachter, "Near-optimal probabilistic RNA-seq quantification," *Nature Biotechnology*, May 2016.

[147] R. Patro, G. Duggal, M. I. Love, R. A. Irizarry, and C. Kingsford, "Salmon provides fast and bias-aware quantification of transcript expression," *Nature Methods*, 2017.

[148] M. Pertea, D. Kim, G. M. Pertea, J. T. Leek, and S. L. Salzberg, "Transcript-level expression analysis of RNA-seq experiments with HISAT, StringTie and Ballgown," *Nature Protocols*, 2016.

[149] C. Trapnell, A. Roberts, L. Goff, G. Pertea, D. Kim, D. R. Kelley, H. Pimentel, S. L. Salzberg, J. L. Rinn, and L. Pachter, "Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks," *Nature Protocols*, 2012.

[150] J. A. Martin and Z. Wang, "Next-generation transcriptome assembly," *Nature Reviews Genetics*, 2011.

[151] G. P. Wagner, K. Kin, and V. J. Lynch, "Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples," *Theory in Biosciences*, 2012.

[152] B. Li, V. Ruotti, R. M. Stewart, J. A. Thomson, and C. N. Dewey, "RNA-Seq gene expression estimation with read mapping uncertainty," *Bioinformatics*, 2010.

[153] A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold, "Mapping and quantifying mammalian transcriptomes by RNA-Seq," *Nature Methods*, 2008.

[154] N. J. Schurch, P. Schofield, M. Gierliński, C. Cole, A. Sherstnev, V. Singh, N. Wrobel, K. Gharbi, G. G. Simpson, T. Owen-Hughes, M. Blaxter, and G. J. Barton, "How many biological replicates are needed in an RNA-seq experiment and which differential expression tool should you use?," *RNA*, 2016.

[155] J. Quackenbush, "Microarray data normalization and transformation," *Nature Genetics*, 2002.

[156] M.-A. Dillies, A. Rau, J. Aubert, C. Hennequet-Antier, M. Jeanmougin, N. Servant, C. Keime, G. Marot, D. Castel, J. Estelle, G. Guernec, B. Jagla, L. Jouneau, D. Laloe, C. Le Gall, B. Schaeffer, S. Le Crom, M. Guedj, and F. Jaffrezic, "A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis," *Briefings in Bioinformatics*, 2013.

[157] A. A. Hernandez-Lopez, J. Voges, C. Alberti, M. Mattavelli, and J. Ostermann, "Lossy Compression of Quality Scores in Differential Gene Expression: A First Assessment and Impact Analysis," in *2018 Data Compression Conference*, (Snowbird, UT), IEEE, 2018.

[158] U. Nagalakshmi, Z. Wang, K. Waern, C. Shou, D. Raha, M. Gerstein, and M. Snyder, "The Transcriptional Landscape of the Yeast Genome Defined by RNA Sequencing," *Science*, 2008.

[159] R. Lister, R. C. O'Malley, J. Tonti-Filippini, B. D. Gregory, C. C. Berry, A. H. Millar, and J. R. Ecker, "Highly Integrated Single-Base Resolution Maps of the Epigenome in Arabidopsis," *Cell*, 2008.

[160] B. T. Wilhelm, S. Marguerat, S. Watt, F. Schubert, V. Wood, I. Goodhead, C. J. Penkett, J. Rogers, and J. BÃ€hler, "Dynamic repertoire of a eukaryotic transcriptome surveyed at single-nucleotide resolution," *Nature*, 2008.

[161] S. Li, P. P. Łabaj, P. Zumbo, P. Sykacek, W. Shi, L. Shi, J. Phan, P.-Y. Wu, M. Wang, C. Wang, D. Thierry-Mieg, J. Thierry-Mieg, D. P. Kreil, and C. E. Mason, "Detecting and correcting systematic variation in large-scale RNA sequencing data," *Nature Biotechnology*, 2014.

[162] "SEQC/MAQC-III Consortium, A comprehensive assessment of RNA-seq accuracy, reproducibility and information content by the Sequencing Quality Control Consortium," *Nature Biotechnology*, 2014.

[163] D. Kim, G. Pertea, C. Trapnell, H. Pimentel, R. Kelley, and S. L. Salzberg, "TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions," *Genome Biology*, 2013.

[164] D. Kim, B. Langmead, and S. L. Salzberg, "HISAT: a fast spliced aligner with low memory requirements," *Nature Methods*, 2015.

[165] K. E. Hayer, A. Pizarro, N. F. Lahens, J. B. Hogenesch, and G. R. Grant, "Benchmark analysis of algorithms for determining and quantifying full-length mRNA splice forms from RNA-seq data," *Bioinformatics*, 2015.

[166] C. Trapnell, B. A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. J. van Baren, S. L. Salzberg, B. J. Wold, and L. Pachter, "Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation," *Nature Biotechnology*, 2010.

[167] M. Pertea, G. M. Pertea, C. M. Antonescu, T.-C. Chang, J. T. Mendell, and S. L. Salzberg, "StringTie enables improved reconstruction of a transcriptome from RNA-seq reads," *Nature Biotechnology*, 2015.

[168] M. Gierliński, C. Cole, P. Schofield, N. J. Schurch, A. Sherstnev, V. Singh, N. Wrobel, K. Gharbi, G. Simpson, T. Owen-Hughes, M. Blaxter, and G. J. Barton, "Statistical models

for RNA-seq data derived from a two-condition 48-replicate experiment," *Bioinformatics*, 2015.

[169] M. I. Love, W. Huber, and S. Anders, "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2," *Genome Biology*, 2014.

[170] T. Griebel, B. Zacher, P. Ribeca, E. Raineri, V. Lacroix, R. Guigó, and M. Sammeth, "Modelling and simulating generic RNA-Seq experiments with the flux simulator," *Nucleic Acids Research*, 2012.

[171] "European Nucleotide Archive. Highly replicated yeast RNAseq; Study: PRJEB5348." [ONLINE] Available at: https://www.ebi.ac.uk/ena/data/view/PRJEB5348.

[172] "European Nucleotide Archive. Homo Sapiens; Study: PRJNA222975." [ONLINE] Available at: https://www.ebi.ac.uk/ena/data/view/PRJNA222975.

[173] J. M. Boer, W. K. Huber, H. SÃŒltmann, F. Wilmer, A. von Heydebreck, S. Haas, B. Korn, B. Gunawan, A. Vente, L. FÃŒzesi, M. Vingron, and A. Poustka, "Identification and Classification of Differentially Expressed Genes in Renal Cell Carcinoma by Expression Profiling on a Global Human 31,500-Element cDNA Array," *Genome Research*, 2001.

[174] F. Seyednasrollah, A. Laiho, and L. L. Elo, "Comparison of software packages for detecting differential expression in RNA-seq studies," *Briefings in Bioinformatics*, 2015.

[175] A. Nekrutenko and J. Taylor, "Next-generation sequencing data interpretation: enhancing reproducibility and accessibility," *Nature Reviews Genetics*, 2012.

[176] A. McKenna, M. Hanna, E. Banks, A. Sivachenko, K. Cibulskis, A. Kernytsky, K. Garimella, D. Altshuler, S. Gabriel, M. Daly, and M. A. DePristo, "The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data," *Genome Research*, 2010.

[177] E. P. Consortium, "An integrated encyclopedia of DNA elements in the human genome," *Nature*, 2012.

[178] K. M. Fisch, T. Meißner, L. Gioia, J.-C. Ducom, T. M. Carland, S. Loguercio, and A. I. Su, "Omics Pipe: a community-based framework for reproducible multi-omics data analysis," *Bioinformatics*, 2015.

[179] B. Fjukstad and L. A. Bongo, "A Review of Scalable Bioinformatics Pipelines," *Data Science and Engineering*, 2017.

[180] E. Smith, "Data Analysis with CASAVA v1.8and the MiSeqReporter," 2011.

[181] F. R. Pinu, D. J. Beale, A. M. Paten, K. Kouremenos, S. Swarup, H. J. Schirra, and D. Wishart, "Systems Biology and Multi-Omics Integration: Viewpoints from the Metabolomics Research Community," *Metabolites*, 2019.

[182] B. L. Cantarel, D. Weaver, N. McNeill, J. Zhang, A. J. Mackey, and J. Reese, "BAYSIC: a Bayesian method for combining sets of genome variants with improved specificity and sensitivity," *BMC Bioinformatics*, 2014.

[183] A. Gézsi, B. Bolgár, P. Marx, P. Sarkozy, C. Szalai, and P. Antal, "VariantMetaCaller: automated fusion of variant calling pipelines for quantitative, precision-based filtering," *BMC Genomics*, 2015.

[184] S. Roy, C. Coldren, A. Karunamurthy, N. S. Kip, E. W. Klee, S. E. Lincoln, A. Leon, M. Pullambhatla, R. L. Temple-Smolkin, K. V. Voelkerding, C. Wang, and A. B. Carter, "Standards and Guidelines for Validating Next-Generation Sequencing Bioinformatics Pipelines: A Joint Recommendation of the Association for Molecular Pathology and the College of American Pathologists," *The Journal of Molecular Diagnostics*, 2018.

[185] R. Norel, J. J. Rice, and G. Stolovitzky, "The self-assessment trap: can we all be better than average?," *Molecular Systems Biology*, 2011.

[186] Y. Diao, A. Roy, and T. Bloom, "Building Highly-Optimized, Low-Latency Pipelines for Genomic Data Analysis," *7th Biennial Conference on Innovative Data Systems Research*, 2015.

[187] H. Li and N. Homer, "A survey of sequence alignment algorithms for next-generation sequencing," *Briefings in Bioinformatics*, 2010.

[188] N. A. Fonseca, J. Rung, A. Brazma, and J. C. Marioni, "Tools for mapping high-throughput sequencing data," *Bioinformatics*, 2012.

[189] C. Notredame, "Recent progress in multiple sequence alignment: a survey," *Pharmacogenomics*, 2002.

[190] D. Earl, N. Nguyen, G. Hickey, R. S. Harris, S. Fitzgerald, K. Beal, I. Seledtsov, V. Molodtsov, B. J. Raney, H. Clawson, J. Kim, C. Kemena, J.-M. Chang, I. Erb, A. Poliakov, M. Hou, J. Herrero, W. J. Kent, V. Solovyev, A. E. Darling, J. Ma, C. Notredame, M. Brudno, I. Dubchak, D. Haussler, and B. Paten, "Alignathon: a competitive assessment of whole-genome alignment methods," *Genome Research*, 2014.

[191] T. Steijger, J. F. Abril, P. G. Engström, F. Kokocinski, T. J. Hubbard, R. Guigó, J. Harrow, and P. Bertone, "Assessment of transcript reconstruction methods for RNA-seq," *Nature Methods*, 2013.

[192] P. Flicek and E. Birney, "Sense from sequence reads: methods for alignment and assembly," *Nature Methods*, 2009.

[193] A. Hatem, D. Bozdağ, A. E. Toland, and Ü. V. Çatalyürek, "Benchmarking short sequence mapping tools," *BMC Bioinformatics*, 2013.

[194] H. Li, "Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM," *arXiv:1303.3997 [q-bio]*, 2013. arXiv: 1303.3997.

[195] S. Thankaswamy-Kosalai, P. Sen, and I. Nookaew, "Evaluation and assessment of read-mapping by multiple next-generation sequencing aligners based on genome-wide characteristics," *Genomics*, 2017.

[196] A. D. Smith, Z. Xuan, and M. Q. Zhang, "Using quality scores and longer reads improves accuracy of Solexa read mapping," *BMC Bioinformatics*, 2008.

[197] H. Li, J. Ruan, and R. Durbin, "Mapping short DNA sequencing reads and calling variants using mapping quality scores," *Genome Research*, 2008.

[198] Novocraft, "Novoalign & NovoalignCS Reference Manual," tech. rep., 2017.

[199] A. A. Hernandez-Lopez, C. Alberti, and M. Mattavelli, "Toward a Dynamic Threshold of Quality-Score Distortion in Reference-Based Alignment," in *15th International Symposium on Bioinformatics Research and Applications*, (Barcelona, Spain), 2019.

[200] A. Hernandez-Lopez, C. Alberti, and M. Mattavelli, "Toward a dynamic threshold for quality-score distortion in reference-based alignment," *Journal of Computational Biology*, (accepted; preprint bioRxiv:10.1101/754614), 2019.

[201] "European Nucleotide Archive. T16M Metastatic liver tumor; File: SRR089705.fastq.gz." [ONLINE] Available at: https://www.ebi.ac.uk/ena/data/view/SRR089705.

[202] "European Nucleotide Archive. Gene expression data in skin fibroblast cells; File: SRR7093809.fastq.gz." [ONLINE] Available at: https://www.ebi.ac.uk/ena/data/view/PRJNA454681.

[203] J. Quackenbush, "Computational analysis of microarray data," *Nature Reviews Genetics*, 2001.

[204] Illumina, "Understanding illumina quality scores," technical Note, 2012.

Ana Hernandez-Lopez

ana.ang.herlop@gmail.com https://orcid.org/0000-0001-6009-7092
+41 78 838 3673 ana.ang.herlop hlana 1015 Lausanne, Switzerland

EDUCATION

Switzerland

PhD in Computer & Communication Sciences Sept. 2014 – Oct. 2019
Swiss Federal Institute of Technology, Lausanne (EPFL)
Research on new approaches to genomic information representation and processing

Mexico

Master of Science in Computer Science 2011 – 2013
Center for Research and Advances Studies of the National Polytechnic Institute of Mexico (CINVESTAV-LTI)
Research on computer vision in embedded systems

Bachelor in Electrical and Electronic Engineering 2002 – 2009
National Autonomous University of Mexico, Faculty of Engineering (UNAM-FI), Institute of Engineering (IIUNAM) *courses: 5 years + thesis: 2 years*
Major in Electronics

CORE EXPERIENCE

Switzerland

SCI-STI-MM group | **EPFL** 2015 – 2019

Doctoral dissertation:

"On the relevance of quality score metadata in genomic sequence data for omics applications"

- Fully responsible to frame and execute the research proposal: identification and assessment of research problems and requirements, design of promising solutions, implementation, analysis and evaluation of experiments, and documentation of results
- Developed workflows to process high-throughput sequence data (whole genome and RNA-seq) for variant calling, differential gene expression, and sequence alignment
- Investigated the impact of lossy compression of genomic metadata in the context of variant calling and differential gene expression pipelines, and sequence alignment
- Surveyed downstream applications and tools for genome sequence analysis, and proposed pipelines to explore the impact of lossy compression on genomic sequences
- Examined and assessed the effect of lossy compression of genomic metadata and published findings

MPEG-G | *MPEG Genomic Compression Group* 2015 – 2018

Swiss delegate to the ISO/IEC JTC 1/SC 29/WG 11 working group (MPEG), Genomic Data Compression

- Investigated genomic information storage and compression to identify problems in the representation and manipulation of genomic data
- Contributed to the documentation of findings and dissemination activities of the group

Swiss Institute of Bioinformatics (SIB/Vital-IT) | *PoSeNoGap Project* 2015 – 2016

Research assistant and IT technical support

- Inspected, analyzed and preprocessed genomic files
- Researched and evaluated genomic data compressors to define a benchmark strategy that provided quantitative evidence of the storage footprint of genomic metadata

Mexico

CINVESTAV-LTI | *Department of Embedded Systems and Reconfigurable Computing* 2014

Research assistant (6 months)

- Optimized the architecture proposed in my master thesis for its use in an hexapod robot
- Documented the results for a scientific publication

ADDITIONAL EXPERIENCE

Mexico

Nextel | *Network Operation Center* 2010

Monitoring engineer

Siemens | *Business Learning Program* 2009

Trainee in the Energy sector, division of Electrical Substations

IIUNAM | *Department of Instrumentation* 2006 – 2009

Research and project assistant

- Implemented and evaluated electronic prototypes based on microcontrollers to instrument systems for academic and industrial projects
- Developed a data recording module based on SD card memories for a portable seismic unit (bachelor thesis). This system spearheaded the design of a multi-purpose seismic data logger, an in-house ongoing project in the seismic instrumentation department at IIUNAM

TECHNICAL SKILLS

Shell scripting: Bash, AWK, Perl

Processing and Visualization: R, tidyverse (tidyr, dplyr, ggplot2, etc), R shiny, Python, SciPy, NumPy, Scikit-learn, Matlab, C, genome browsers and viewers

Others: SAMtools, bcftools, GATK, Illumina BaseSpace, Galaxy, TeX/LaTeX, Git, UNIX/Linux OS

Experimenting with: knitr, Stan, Qiskit, Cirq

RESEARCH OUTPUTS

PUBLICATIONS

Hernandez-Lopez, A., Alberti, C. and Mattavelli, M. "Toward a Dynamic Threshold for Quality-Score Distortion in Reference-Based Alignment." **Journal of Computational Biology.** (Accepted; preprint: DOI: 10.1101/754614)

Hernandez-Lopez, A., Voges, J., Alberti, C., Mattavelli, M. and Ostermann, J. "Lossy Compression of Quality Scores in Differential Gene Expression: A first Assessment and Impact Analysis." **IEEE 2018 Data Compression Conference**, Snowbird, Utah, USA, March 27–29, 2018. DOI: 10.1109/DCC.2018.00025

Alberti, C., Daniels, N., Hernaez, M., Voges, J., Goldfeder, R., Hernandez-Lopez, A., Mattavelli, M. and Berger, B. "An Evaluation Framework for Lossy Compression of Genome Sequencing Quality Values." **IEEE 2016 Data Compression Conference**, Snowbird, Utah, USA, March 29-April 1, 2016. DOI:10.1109/DCC.2016.39

Hernandez-Lopez, A., Torres-Huitzil, C. and Garcia-Hernandez, JJ. "FPGA-Based Flexible Hardware Architecture for Image Interest Point Detection." **International Journal of Advanced Robotic Systems (IJARS)**, 2015. (IJARS Women in Robotics winning paper) DOI: 10.5772/61058

PRESENTATIONS

"Relevance of QVS Information for Analysis Applications of Genomic Sequencing Data." **IEEE Data Science Workshop**, Minneapolis, MN, USA, June 2–5, 2019.

"Toward a Dynamic Threshold for Quality-Score Distortion in Reference-Based Alignment." **15th International Symposium on Bioinformatics Research and Applications (ISBRA)**, Barcelona, Spain, June 3–6, 2019.

OTHERS

"Differential Gene Expression with Lossy Compression of Quality Scores in RNA-Seq Data." **IEEE 2017 Data Compression Conference**, Snowbird, Utah, USA, April 4–7, 2017. (Abstract, DOI: 10.1109/DCC.2017.75; Poster)

"Transcriptome reconstruction with quality score distortion in reference-based alignment." **Research in Computational Molecular Biology (RECOMB)**, Paris, France, April 19–24, 2018. (Poster)

LANGUAGES

English	Fluent spoken and written (C2)
French	Fluent spoken and written (C1)
Spanish	Native language

HONORS AND AWARDS

EDIC fellowship at EPFL	2014
CONACYT postgraduate scholarship	2011 – 2013
IIUNAM undergraduate scholarship	2007 – 2009

EXTRACURRICULAR ACTIVITIES

Avid reader of general psychology and existential philosophy. Also reader of The Economist, Quanta magazine and Le Monde

Coffee enthusiast: roasting specialty coffee beans, pour-over brewing and lever espresso extraction

Plant-based cooking: exploring flavor profiles with minimal ingredients

Long-distance running: 10 km and half marathon