Exploring Lossy Compression of Gene Expression Matrices

Coleman B. McKnight*, Alexandra L Poulos†, M. Reed Bender‡, Jon C. Calhoun†, F. Alex Feltus*
* Department Genetics and Biochemistry, Clemson University, Clemson, South Carolina 29634
† Holcombe Department of Electrical and Computer Engineering,
Clemson University, Clemson, South Carolina 29634
‡ Biological Data Science and Informatics, Clemson University, Clemson, South Carolina 29634
{cbmckni, alpoulo, mrbende, jonccal, ffeltus}@clemson.edu

***Abstract*—Gene Expression Matrices (GEMs) are a fundamental data type in the genomics domain. As the size and scope of genomics experiments increase, researchers are struggling to process large GEMs through downstream workflows with currently accepted practices. In this paper, we propose a methodology to reduce the size of GEMs using multiple approaches. Our method partitions data into discrete fields based on data type and employs state-of-the-art lossless and lossy compression algorithms to reduce the input data size. This work explores a variety of lossless and lossy compression methods to determine which methods work the best for each component of a GEM. We evaluate the accuracy of the compressed GEMs by running them through the Knowledge Independent Network Construction (KINC) workflow and comparing the quality of the resulting gene co-expression network with a lossless control to verify result fidelity. Results show that utilizing a combination of lossy and lossless compression results in compression ratios up to 9.77× on a Yeast GEM, while still preserving the biological integrity of the data. Usage of the compression methodology on the Cancer Cell Line Encyclopedia(CCLE) GEM resulted in compression ratios up to 9.26×. By using this methodology, researchers in the Genomics domain may be able to process previously inaccessible GEMs while realizing significant reduction in computational costs.**

***Index Terms*—lossy compression, gene expression matrices, genomics, gene co-expression networks, RNAseq**

I. Introduction

A Gene Expression Matrix (GEM) is a data structure that contains comprehensive gene expression quantification for m genes across n biological samples. GEMs are commonly used in the genomics discipline and have been the source of significant findings since the commercialization of high-density microarrays in the 1990s [1]. GEMs are used as input to various scientific workflows such as differential gene expression [2], [3] and gene co-expression network (GCN) analysis [4], [5]. Due to recent technological advancements in cyberinfrastructure [6]–[9] and DNA sequencing technology [10], the accumulation of RNA expression data-sets is geometric leading to larger GEMs for thousands of species. It is becoming routine to process biomedical GEMs with 50-80 thousand genes and 10-20 thousand samples. In the near future, it is conceivable that GEMs could swell to millions of samples and 200 thousand gene products. Although GEMs are able to fit into most modern storage systems, complex matrix operations in downstream workflows render many GEMs too large for processing in memory. Even if the processing of a large GEM is made possible, the required computational and monetary resources are tangible and need to be controlled.

Compression is a standard practice to reduce data set size. Lossless compression preserves all of the original data, ensuring that no information is lost. For floating-point data-sets, the compression ratio of lossless compression is limited by its guarantee of exact accuracy [11]. One solution to improve the compression ratio is lossy compression. Lossy compression trades the loss of precision within a certain error bound for compression ratios than can be more than an order-of-magnitude more than the best lossless compression methods [12]. With lossy compression, the compressabilty of a data-set is dependant on the selection of error bound and error bounding metric, and selection is often application dependant.

Given the increasing difficulty to process GEMs through downstream workflows, and the potential benefits offered by data compression, we introduce the concept of lossless and lossy compression of GEMs. Compression of GEMs is intended to reduce the size of the data while still maintaining its integrity. This paper presents a methodology of compression GEMs and serves as a first step to integrate in-line lossy compression into genomic workflows. Integrating lossy compression into genomic workflows enables researchers to process previously inaccessible GEMs and realize significant reduction in resource costs. This paper makes the following contributions:

- the first application of lossy compression on GEMs, including a representative comparison on the performance of GEM compression with state-of-the-art compression methods; and
- a methodology for compressing GEMs that still maintains the biological integrity of the data.

II. Background

A. Data Compression

To mitigate bandwidth and storage bottlenecks, HPC applications employ a variety of data reduction techniques. Decimation saves I/O bandwidth and capacity by storing data from a subset of the simulation's time-steps. Decreasing the number of time-steps logged for analysis diminishes the ability

978-1-7281-6017-7/19/$31.00 ©2019 IEEE
DOI 10.1109/DRBSD-549595.2019.00010

to conduct meaningful scientific research. Data compression techniques reduce the per time-step data size, enabling more time-steps to be saved for analysis. Data compression techniques fall into one of two classifications: lossless and lossy. Lossless compressors such as FPC [13], fp-zip [14], Gzip [15], and zstd [16] reduce the data set's size without impacting the accuracy of the data. However, for data coming from HPC applications, most achieve compression ratios between 1–4×. Lossy data compression is able to trade accuracy in the decompressed data for larger reductions in data set size. Truncating data from 64-bit precision to 32-bit precision results in a compression ratio of 2×. Specially designed lossy compression algorithms such as SZ [12] and ZFP [17] enable compression ratios of 10× or more while at the same time bounding the error introduced on a per element or a per data array basis. Although lossy compression results in large reductions in data size, setting the compressor's error bound is often domain specific [18]–[20]. Understanding the best error bound and error bounding mode for particular disciplines remains an open question.

B. Gene Expression Matrix (GEM)

Modern high-throughput DNA sequencing technology has dramatically increased the resolution of observations into biological systems at the molecular level. It is now routine for biological research labs to re-sequence the DNA of whole chromosomes from an individual organism to identify stable DNA sequence differences (DNA polymorphisms) between individuals to determine the interactions leading to alternate expression of a trait. DNA polymorphisms, stored in VCF-format [21], can be inserted into genotype matrices that coded as integers representing the specific sequence differences at (chromosome, start, stop) coordinates on a reference genome scaffold for any number of individual genomes. High-throughput DNA sequencing experiments answer critical questions in biology and are being deposited in repositories at a rapid pace opening the door for mixing experiments and mining the datasets for new insight. The National Center for Biotechnology Information (NCBI) Sequence Read Archive (SRA) database [22] contains over 30 petabytes of raw DNA sequence datasets from thousands of species.

In addition to the measurement of stable DNA polymorphisms between organisms, life scientists measure dynamic gene expression in a cell, tissue, or organ. In a typical experiment, control samples are compared to similar samples presented with an alternate condition. For example, normal kidney tissue can be compared to tumor tissue from the same patient [23]; plant roots can be compared between normal or reduced fertilizer conditions [23], [24]. Samples are processed using wet-lab molecular biology techniques to extract RNA from all samples, RNA is converted to cDNA, and the cDNA molecules are sequenced at the scale of tens of millions of DNA sequence reads and stored in FASTQ-format files [25]. In the dry-lab phase of the experiment, individual cDNA sequence reads are aligned to a reference genome of the target species to create a SAM file [26].

The expression level of each of thousands of genes for each sample are quantified and stored in floating-point Gene Expression Matrix (GEM). A representative computational workflow to prepare a GEM can be found at [27]. GEMs are a fundamental datatype for downstream analysis of gene expression relationships between biological conditions. For example, all of the thousands of genes in the GEM can be analyzed for pairwise correlation within specific conditions — e.g., Knowledge Independent Network Construction (KINC) algorithm [5]. Alternatively, genes can be tested for statistically significant differential expression between conditions — e.g. DEseq2 algorithm [2]. Our group mines a GEM from the Genotype-Tissue Expression (GTEx) project [28] comprising of gene expression measurements for 56,202 human genes across 11,688 samples representing 53 human tissue types. We also mine a GEM from The Cancer Genome Atlas (TCGA) [29] that contains 60,101 gene expression measurements across 11,093 tumor and normal tissue representing 33 types of cancer. As DNA sequencing costs fall and data repositories fill with useful datasets, GEMs consume considerable amount of storage, posing new computational challenges. In this paper, we describe computational optimization techniques focused on real gene expression quantification data stored in floating point GEMs with ASCII metadata — i.e., gene and sample identifiers.

C. Knowledge Independent Network Construction (KINC)

Biological noise is introduced from natural, systematic and statistical variation sources [5], [30]–[32]. Gene expression, for example, is intrinsically noisy within a cell or tissue and further systemic noise is introduced during the measurement process. When the cellular conditions change due to developmental or environmental perturbations of the system, extrinsic noise is added. Further the choice of test and normalization statistics and their algorithm implementations introduce noise during the processing of raw RNAseq or microarray measurements created by selection of data processing tools. We use the Knowledge Independent Network Construction (KINC) gene co-expression network (GCN) construction software to reduce natural extrinsic condition-specific noise by clustering samples prior to gene correlation analysis [5].

In GEMs, gene expression noise is present in the floating point values that quantify the expression of a particular gene in each sample. A fundamental challenge present in any type of computational science is determining how much of the precision of generated data is actually significant and not just random noise. For GEMs, there is currently no standard for how much precision can be discarded. Lossy compression methods produce better compression ratios when more of the original data is able to be discarded, so finding the optimal degree of compression for GEMs is of interest. The optimal degree of compression for GEMs is defined as the error bound of a lossy compressor that results in the best compression ratio for a GEM while still maintaining the biological integrity of the data by comparing a KINC GCN built from a compressed

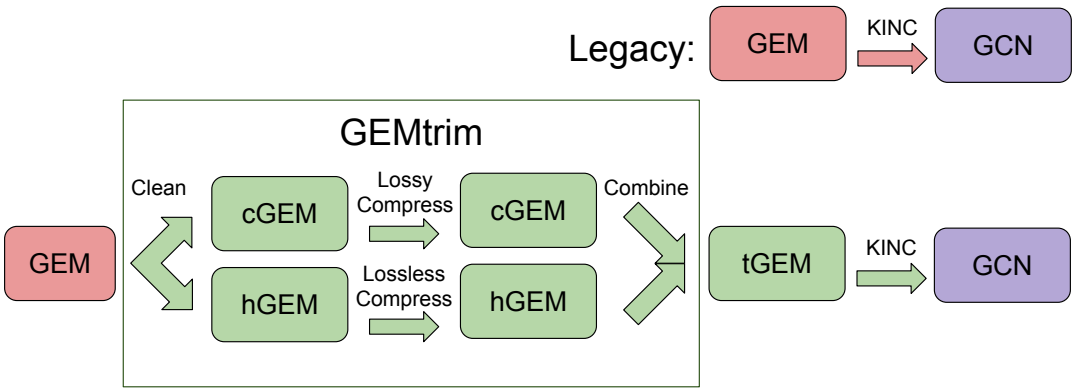

Fig. 1. Use of the GEMtrim methodology with KINC compared to the legacy practice of GCN creation.

GEM to one that was constructed with KINC without compression.

III. Related Work

Lossy compression algorithms have seen rapid development recently. Work has been done to improve the performance of lossy compression algorithms for HPC [33], [34], provide more error bounding metrics such as PSNR [35] and point-wise relative error [36]. As compressors develop, researches have explore how to integrate them into HPC applications and hte impact or accuracy [18]–[20], [37], [38].

Many methodologies have been proposed to reduce the size of genomic data in various stages of the sequencing and analysis pipeline. Several lossless compression methods are used to compress raw (FASTQ) and aligned and/or pseudoaligned (SAM/BAM) high-throughput sequenced data [39]–[43]. In some workflows, lossy compression is used on the entirety of the sequenced data, while other approaches only use lossy compression on specific parts of FASTQ files, such as quality values [40], [42], [44], [45]. These data are then used as input to another process in the pipeline, for example, Hisat2 [46], Kallisto [47], or Salmon [48] to create a GEM file for further analysis. To the authors' knowledge, our work exploring the application of lossy compression to GEM files is the first of its kind.

IV. Compressing GEMs

A. Overview

This section describes a methodology for the compression of GEMs for downstream processing. This methodology represents the most viable option that is available to researchers that intend to compress GEMs. We introduce GEMtrim, a methodology that converts state-of-the practice GEM files into a compressed trimmed GEM (tGEM):

1) The separation of textual and floating-point data from the GEM.
2) Lossless compression of the binary textual data.
3) Lossy compression of the binary floating-point data.
4) Recombination of the floating-point and binary data into the final tGEM file.

Figure 1 visualizes this process. The standard "Legacy" process of GCN production shows a raw GEM (shown in red) is used as input for the KINC workflow to produce a GCN. With the proposed GEMtrim methodology (highlighted in green), the GEM is split into textual (hGEM) and floating-point (cGEM) data. The textual data is compressed using a lossless method while lossy compression is utilized for the floating-point data. The compressed headers and floating-point data are combined into a single binary format, creating a tGEM(shown in green). The tGEM is used as input by KINC or any other compatible workflow. Currently KINC does not accept tGEM inputs; therefore, we convert tGEMs to standard GEMs for the purposes of this study. Future work will alleviate this conversion by adding a tGEM reader to KINC.

B. Description

The GEM data structure is composed of a $m \times n$ heterogeneous matrix of floating-point and textual values, prepended with a row of sample headers and a column of gene headers. When a gene is not expressed in a particular sample, the value is replaced with a "NA" value to denote its lack of expression. To compress GEMs, the textual data within the GEM has to be isolated from the floating-point values of gene expression. The headers are removed from the GEM and compressed separately in a lossless manner, then stored for reinsertion downstream. The "NA" values are replaced with floating-point values of "0.0", with their locations being stored for reinsertion downstream. The resulting two dimensional matrix of floating-point values is then converted from the original ASCII format to a binary file. This binary data is designated as a *cleaned* version of the original GEM, and is then passed as input for the compression step. Textual data is sent to lossless compression algorithms. The floating-point data is sent to either a lossless or lossy compressor. After compression, the binary data is combined to into a trimmed GEM (tGEM) that is suitable as KINC input. The tGEM is stored as a compressed binary version of the raw GEM, and can be used as input for any compatible workflow.

V. Experimental Results

A. Experimental Design

To determine how much precision of a GEM is required to reproduce a GCN, we evaluate how lossy compression of tGEMs derived from real RNA-seq data impacts the fidelity of KINC. This section describes the steps taken to produce the a methodology for the compression of GEMs, while still maintaining the biological integrity of the data that is needed for downstream processing.

For evaluation, we use a yeast GEM comparing the RNA expression of 7,050 genes across 188 yeast samples. This small GEM is used for initial testing to allow for practical downstream processing through the KINC (v3.2.2) workflow, which would need to be done for each compression method used. In addition, we explore a larger GEM consisting of 56,202 genes across 1,019 human cancer cell lines (Cancer Cell Line Encyclopedia (CCLE) [49]).

To convert and compress GEMs into tGEMs, we explore a variety of lossless and lossy compression methods representative of the methods currently available in an HPC environment. In particular, we explore zSTD (v1.4.3) [16], zLib (v1.2.11), gZip (v1.5.0) [15], bZip2 (v1.0.6) [50], fpzip (v1.2.0) [14], and the lossless mode of ZFP (v0.5.5) [17] for lossless compression. For lossy compression, we use SZ (v2.0.2.0) [34] and ZFP (v0.5.5) [17] and a variety of error bounds and error bounding types.

All compression are done on Clemson's Palmetto Cluster, using a single node with an Intel Xeon Gold 6148 CPU.

B. Yeast

The yeast GEM was successfully compressed using each configuration. Although we test 40 different compression configurations, only data for those that passed KINC validation are displayed.

Figure 2 shows the compression ratio of the final compressed tGEM file(headers and expression values) for each compressor and compression configuration. The highest compression ratio of 9.77× was produced using SZ with the Peak Signal-to-Noise Ratio (PSNR) error bound set to 100, followed by 9.19× with the Point-Wise Relative (PWREL) error bound set to $1e{-}4$. The lossless compressors all resulted in ratios around 4.8×, with the exception of fp-zip at 5.45× and the lossless mode of ZFP at 4.16×.

Figure 3 shows the compression and decompression bandwidth (MB/s) of each configuration. The best lossy compression bandwidth was achieved by SZ using the PSNR error bound set to 100. While the lossless compressors had similar performance in compression ratio, they varied on compression bandwidth, with zLib outperforming and bZip2 underperforming the rest. The best decompression bandwidth was achieved by SZ using the PSNR error bound set to 100. The lossless compressors differed greatly when decompressing, with with fp-zip outperforming and zSTD under-performing the rest.

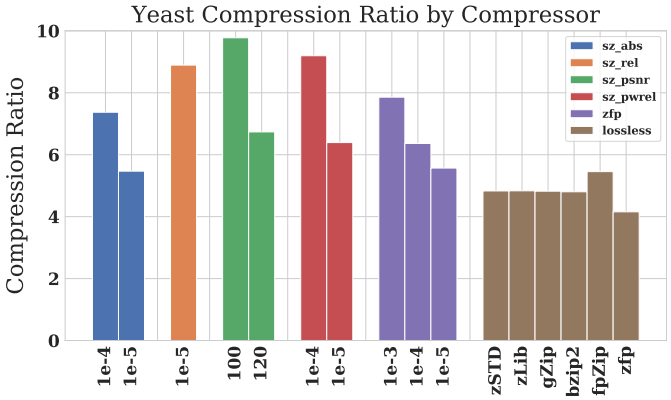

Fig. 2. Compression ratio of lossless and lossy compression methods on yeast data.

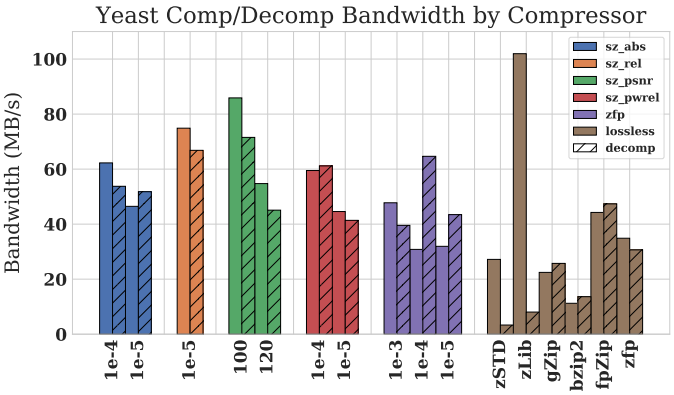

Fig. 3. Compression and Decompression bandwidth of lossless and lossy compression methods on yeast data.

1) Accuracy: To verify accuracy of the tGEM, the resulting binary data was then read into a script that reinserted the sample and gene headers as the first row and column and then reinserted each "NA" value in its original location. In order to prevent altering KINC source code, the resulting GEM was then written back into the original ASCII format needed for use as input. Future work will add a tGEM reading module to KINC. Each GEM is sent as input for the KINC workflow. KINC's output is then compared to a reference GCN using a uncompressed GEM.

We define two metrics as significant for comparison of GCNs: significance threshold and the number of edges in the network. The significance threshold value is the Spearman correlation cut-off determined by Random Matrix Theory [5] which builds a biologically realistic scale-free gene co-expression graph. The number of edges represent the number of significant gene correlations that are assumed to be biologically relevant co-functional genetic relationships. If the significance threshold and number of edges from a lossy KINC run is identical to the lossless KINC run used as a control, the network and the lossy compression method used are designated as *valid*.

Of the 34 lossy compression configurations tested, only

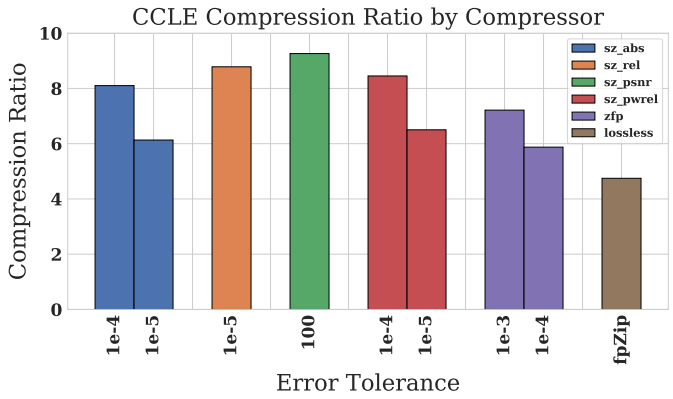

Fig. 4. Compression ratios of lossless and lossy compression methods on CCLE data.

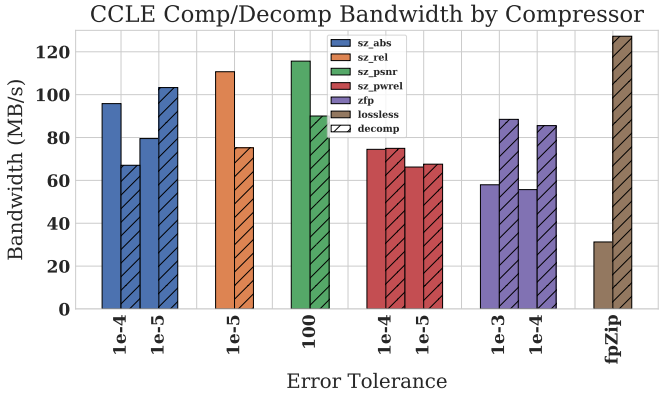

Fig. 5. Compression and decompression bandwidth of lossless and lossy compression methods on CCLE data.

10 passed KINC validation. All 6 lossless methods passed, as the data is identical to the original GEM. The validation described above only allows GCNs with identical thresholds values and edges to pass. The GCN produced using the GEM compressed by SZ using the PSNR bound set to 90 has the correct threshold and only one less edge(2964) then the control(2965), but failed validation. Although the GCNs were essentially the same, further work must be done to determine how much error between GCNs is acceptable for validation.

C. Cancer Cell Line Encyclopedia (CCLE)

Compared to the yeast GEM, the CCLE GEM [49] is substantially larger. This GEM consists of 56,202 genes across 1,019 human cancer cell lines. From the results of Section V-B, we select the configurations that yield valid results to compress and validate the CCLE GEM. The compression ratios and compression/decompression bandwidths for the selected configurations are shown in Figures 4 and 5.

Figure 4 shows the compression ratio of the final compressed tGEM file(headers and expression values) for each compressor and compression configuration. The highest compression ratio of 9.26× was produced using SZ with the PSNR error bound set to 100, followed by 8.78× with the Relative (REL) error bound set to $1e{-}5$. The only lossless compressor tested on the CCLE GEM was fp-zip, as it had the best performance on the yeast GEM. Compression with fp-zip resulted in a ratio around 4.75×.

Figure 5 shows the compression and decompression bandwidth (MB/s) of each configuration. The best lossy compression bandwidth was achieved by SZ using the PSNR error bound set to 100. The best lossy decompression bandwidth was achieved by SZ using the PSNR error bound set to 100. The one lossless compressor had the slowest decompression bandwidth, yet had the highest compression bandwidth tested.

PSNR is the best performing error bound tested in terms of compression ratio and lossy compression bandwidth. These results remain constant for the yeast and CCLE GEMs. PSNR will be a major focus of future work that attempts to estimate the ideal error bound value based the size of a given GEM.

VI. Conclusion and Future Work

As the number and size of GEMs increase, it becomes difficult to process the in downstream workflows. Researchers from various backgrounds will benefit from data reduction techniques that were previously unnecessary. This paper represents the first application of state-of-the-art compression methods to a GEM, a fundamental data structure used in genomics. We devise a methodology known as GEMtrim to convert state-of-the-practice GEMs to versions with lower memory requirements. GEMtrim employs both lossy and lossless compression. Results show that utilizing GEMtrim lead to compression ratios up to 9.77× on a yeast GEM, while still preserving the biological integrity of the data. Usage of this compression methodology on a larger CCLE GEM resulted in compression ratios up to 9.26×.

This work represents the first step toward in-line lossy compression in genomics workflows. Our future work plans to explore the compression of temporary computations in KINC to lower its memory footprint. Reducing the memory footprint of input files and of the application enables researchers to process previously inaccessible GEMs and realize significant reduction in resource costs.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grants No. SHF-1910197 and CC*-1659300.

References

[1] R. Bumgarner, "Overview of dna microarrays: types, applications, and their future," *Current protocols in molecular biology*, vol. 101, no. 1, pp. 22–1, 2013.

[2] M. I. Love, W. Huber, and S. Anders, "Moderated estimation of fold change and dispersion for rna-seq data with deseq2," *Genome biology*, vol. 15, no. 12, p. 550, 2014.

[3] G. K. Smyth, "Limma: linear models for microarray data," in *Bioinformatics and computational biology solutions using R and Bioconductor*. Springer, 2005, pp. 397–420.

[4] P. Langfelder and S. Horvath, "Wgcna: an r package for weighted correlation network analysis," *BMC bioinformatics*, vol. 9, no. 1, p. 559, 2008.

[5] S. P. Ficklin, L. J. Dunwoodie, W. L. Poehlman, C. Watson, K. E. Roche, and F. A. Feltus, "Discovering Condition-Specific Gene Co-Expression Patterns Using Gaussian Mixture Models : A Cancer Case Study," *Scientific Reports*, no. April, pp. 1–11, 2017. [Online]. Available: http://dx.doi.org/10.1038/s41598-017-09094-4

[6] L. Papageorgiou, P. Eleni, S. Raftopoulou, M. Mantaiou, V. Megalooikonomou, and D. Vlachakis, "Genomic big data hitting the storage bottleneck," *EMBnet. journal*, vol. 24, 2018.

[7] 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*, vol. 13, no. 7, p. e1002195, 2015.

[8] N. Mills, E. M. Bensman, W. L. Poehlman, W. B. Ligon III, and F. A. Feltus, "Moving just enough deep sequencing data to get the job done," *Bioinformatics and Biology Insights*, vol. 13, p. 1177932219856359, 2019.

[9] F. A. Feltus, J. R. Breen III, J. Deng, R. S. Izard, C. A. Konger, W. B. Ligon III, D. Preuss, and K.-C. Wang, "The widening gulf between genomics data generation and consumption: A practical guide to big data transfer technology," *Bioinformatics and biology insights*, vol. 9, pp. BBI–S28 988, 2015.

[10] Z. Wang, M. Gerstein, and M. Snyder, "Rna-seq: a revolutionary tool for transcriptomics," *Nature reviews genetics*, vol. 10, no. 1, p. 57, 2009.

[11] S. W. Son, Z. Chen, W. Hendrix, A. Agrawal, W. keng Liao, and A. Choudhary, "Data compression for the exascale computing era - survey," *Supercomputing frontiers and innovations*, vol. 1, no. 2, 2014. [Online]. Available: http://superfri.org/superfri/article/view/13

[12] S. Di and F. Cappello, "Fast error-bounded lossy HPC data compression with SZ," in *2016 IEEE International Parallel and Distributed Processing Symposium, IPDPS 2016, Chicago, IL, USA, May 23-27, 2016*, 2016, pp. 730–739. [Online]. Available: https://doi.org/10.1109/IPDPS.2016.11

[13] M. Burtscher and P. Ratanaworabhan, "High throughput compression of double-precision floating-point data," in *Data Compression Conference, 2007. DCC '07*, March 2007, pp. 293–302.

[14] P. Lindstrom and M. Isenburg, "Fast and efficient compression of floating-point data," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 5, pp. 1245–1250, 2006. [Online]. Available: http://dx.doi.org/10.1109/tvcg.2006.143

[15] P. Deutsch, "Gzip file format specification version 4.3," Aladdin Enterprises, United States, Tech. Rep., 1996.

[16] Y. Collet and M. Kucherawy, "Zstandard Compression and the application/zstd Media Type," RFC 8478, Oct. 2018. [Online]. Available: https://rfc-editor.org/rfc/rfc8478.txt

[17] P. Lindstrom, "Fixed-rate compressed floating-point arrays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2674–2683, Dec 2014.

[18] J. Calhoun, F. Cappello, L. N. Olson, M. Snir, and W. D. Gropp, "Exploring the feasibility of lossy compression for pde simulations," *The International Journal of High Performance Computing Applications*, vol. 33, no. 2, pp. 397–410, 2019. [Online]. Available: https://doi.org/10.1177/1094342018762036

[19] J. Nardi, N. Feldman, A. Poppick, A. Baker, and D. Hammerling, "Statistical analysis of compressed climate data," NCAR, Tech. Rep., 2018.

[20] A. H. Baker, H. Xu, J. M. Dennis, M. N. Levy, D. Nychka, S. A. Mickelson, J. Edwards, M. Vertenstein, and A. Wegener, "A methodology for evaluating the impact of data compression on climate simulation data," in *Proceedings of the 23rd International Symposium on High-performance Parallel and Distributed Computing*, ser. HPDC '14. New York, NY, USA: ACM, 2014, pp. 203–214. [Online]. Available: http://doi.acm.org/10.1145/2600212.2600217

[21] Samtools. The variant call format (vcf) version 4.2 specification. [Online]. Available: https://samtools.github.io/hts-specs/VCFv4.2.pdf

[22] NCBI. Sra database. [Online]. Available: https://www.ncbi.nlm.nih.gov/sra

[23] W. L. Poehlman, J. J. Hsieh, and F. A. Feltus, "Linking binary gene relationships to drivers of renal cell carcinoma reveals convergent function in alternate tumor progression paths," *Scientific reports*, vol. 9, no. 1, p. 2899, 2019.

[24] B. Lagunas, M. Achom, R. Bonyadi-Pour, A. J. Pardal, B. L. Richmond, C. Sergaki, S. Vázquez, P. Schäfer, S. Ott, J. Hammond *et al.*, "Regulation of resource partitioning coordinates nitrogen and rhizobia responses and autoregulation of nodulation in medicago truncatula," *Molecular plant*, vol. 12, no. 6, pp. 833–846, 2019.

[25] SourceForge. Fastq format specification. [Online]. Available: http://maq.sourceforge.net/fastq.shtml

[26] T. S. F. S. W. Group. Sequence alignment/map format specification. [Online]. Available: https://samtools.github.io/hts-specs/SAMv1.pdf

[27] SystemsGenetics. Gemmaker. [Online]. Available: https://github.com/SystemsGenetics/GEMmaker

[28] J. Lonsdale, J. Thomas, M. Salvatore, R. Phillips, E. Lo, S. Shad, R. Hasz, G. Walters, F. Garcia, N. Young *et al.*, "The genotype-tissue expression (gtex) project," *Nature genetics*, vol. 45, no. 6, p. 580, 2013.

[29] K. A. Hoadley, C. Yau, T. Hinoue, D. M. Wolf, A. J. Lazar, E. Drill, R. Shen, A. M. Taylor, A. D. Cherniack, V. Thorsson *et al.*, "Cell-of-origin patterns dominate the molecular classification of 10,000 tumors from 33 types of cancer," *Cell*, vol. 173, no. 2, pp. 291–304, 2018.

[30] Z. Wang and J. Zhang, "Impact of gene expression noise on organismal fitness and the efficacy of natural selection," *Proceedings of the National Academy of Sciences*, vol. 108, no. 16, pp. E67–E76, 2011.

[31] J. M. Raser and E. K. O'Shea, "Noise in gene expression: origins, consequences, and control," *Science*, vol. 309, no. 5743, pp. 2010–2013, 2005.

[32] A. Singh and M. Soltani, "Quantifying intrinsic and extrinsic variability in stochastic gene expression models," *Plos one*, vol. 8, no. 12, p. e84301, 2013.

[33] D. Tao, S. Di, Z. Chen, and F. Cappello, "Significantly improving lossy compression for scientific data sets based on multidimensional prediction and error-controlled quantization," in *2017 IEEE International Parallel and Distributed Processing Symposium, IPDPS 2017, Orlando, FL, USA, May 29 - June 2, 2017*, 2017, pp. 1129–1139. [Online]. Available: https://doi.org/10.1109/IPDPS.2017.115

[34] X. Liang, S. Di, D. Tao, S. Li, S. Li, H. Guo, Z. Chen, and F. Cappello, "Error-controlled lossy compression optimized for high compression ratios of scientific datasets," in *IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10-13, 2018*, 2018, pp. 438–447. [Online]. Available: https://doi.org/10.1109/BigData.2018.8622520

[35] D. Tao, S. Di, X. Liang, Z. Chen, and F. Cappello, "Fixed-psnr lossy compression for scientific data," in *IEEE International Conference on Cluster Computing, CLUSTER 2018, Belfast, UK, September 10-13, 2018*, 2018, pp. 314–318. [Online]. Available: https://doi.org/10.1109/CLUSTER.2018.00048

[36] X. Liang, S. Di, D. Tao, Z. Chen, and F. Cappello, "An efficient transformation scheme for lossy data compression with point-wise relative error bound," in *IEEE International Conference on Cluster Computing, CLUSTER 2018, Belfast, UK, September 10-13, 2018*, 2018, pp. 179–189. [Online]. Available: https://doi.org/10.1109/CLUSTER.2018.00036

[37] D. Laney, S. Langer, C. Weber, P. Lindstrom, and A. Wegener, "Assessing the effects of data compression in simulations using physically motivated metrics," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, ser. SC '13. New York, NY, USA: ACM, 2013, pp. 76:1–76:12. [Online]. Available: http://doi.acm.org/10.1145/2503210.2503283

[38] J. Diffenderfer, A. Fox, J. Hittinger, G. Sanders, and P. Lindstrom, "Error analysis of zfp compression for floating-point data," *SIAM Journal on Scientific Computing*, 02 2019.

[39] M. C. Brandon, D. C. Wallace, and P. Baldi, "Data structures and compression algorithms for genomic sequence data," *Bioinformatics*, vol. 25, no. 14, pp. 1731–1738, 05 2009. [Online]. Available: https://doi.org/10.1093/bioinformatics/btp319

[40] S. Deorowicz and S. Grabowski, "Compression of DNA sequence reads in FASTQ format," *Bioinformatics*, vol. 27, no. 6, pp. 860–862, 01 2011. [Online]. Available: https://doi.org/10.1093/bioinformatics/btr014

[41] 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*, vol. 13, pp. 1005–1008, 2016.

[42] M. Fritz, R. Leinonen, G. Cochrane, and E. Birney, "Efficient storage of high throughput dna sequencing data using reference-based compression," *Genome research*, vol. 21, pp. 734–40, 05 2011.

[43] X. Xia, "Arsda: A new approach for storing, transmitting and analyzing transcriptomic data," *G3-Genes Genomes Genetics*, vol. 7, p. g3.300271.2017, 10 2017.

[44] G. Malysa, M. Hernaez, I. Ochoa, M. Rao, K. Ganesan, and T. Weissman, "QVZ: lossy compression of quality values," *Bioinformatics*, vol. 31, no. 19, pp. 3122–3129, 05 2015. [Online]. Available: https://doi.org/10.1093/bioinformatics/btv330

[45] 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*, vol. 14, p. 187, 06 2013.

[46] G. Wen, "A simple process of rna-sequence analyses by hisat2, htseq and deseq2," in *Proceedings of the 2017 International Conference on Biomedical Engineering and Bioinformatics*, ser. ICBEB 2017. New York, NY, USA: ACM, 2017, pp. 11–15. [Online]. Available: http://doi.acm.org/10.1145/3143344.3143354

[47] N. Bray, H. Pimentel, P. Melsted, and L. Pachter, "Near-optimal probabilistic rna-seq quantification," *Nature Biotechnology*, vol. 34, 04 2016.

[48] A. Srivastava, L. Malik, M. Zakeri, H. Sarkar, C. Soneson, M. Love, C. Kingsford, and R. Patro, "Alignment and mapping methodology influence transcript abundance estimation," 06 2019.

[49] J. Barretina, G. Caponigro, N. Stransky, K. Venkatesan, A. A. Margolin, S. Kim, C. J. Wilson, J. Lehár, G. V. Kryukov, D. Sonkin *et al.*, "The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity," *Nature*, vol. 483, no. 7391, p. 603, 2012.

[50] J. Seward, "bzip2 and libbzip2," *avaliable at http://www. bzip. org*, 1996.