Better quality score compression through sequence-based quality smoothing

Yoshihiro Shibuya(1)† , Matteo Comin(1)

Department of Information Engineering, University of Padova, Padova, Italy

† Email: comin@dei.unipd.it

Motivation
Current NGS techniques are becoming exponentially cheaper. As a result, there is an exponential growth of genomic data unfortunately not followed by an exponential growth of storage, leading to the necessity of compression. Most of the entropy of NGS data lies in the quality values associated to each read, whereas the DNA sequence is highly compressible. Quality values are often more diversified than necessary. Because of that many tools, e.g. Quartz (Nat. Biotech. 2015), try to change (smooth) quality scores in order to improve compressibility without altering the important information they carry for downstream analysis like SNP calling.
Methods
We use the FM-Index, a type of compressed suffix array, to reduce the storage requirements of a dictionary of important k-mers and a simple and effective algorithm to smooth quality values based on the dictionary while maintaining high precision for SNP calling pipelines.
Our algorithm is based on the assumption that every time a reads is aligned to a reference genome the quality values of mismatching positions, e.g. putative SNPs, are more informative than quality value of matches, e.g. conserved bases. Instead of expensive alignment procedure, we consider a reads as the set of its constituent k-mers and we search these k-mers in a dictionary.
If a k-mer is found in the dictionary with a mismatch, most likely the relative base is either an error or a SNP. Both outcomes are best coped with by leaving the quality value of the mismatch untouched. In practice this is obtained by skipping all quality values that corresponds to mismatches in at least one of the k-mers covering a nucleotide, whereas the quality values of matching positions can be smoothed.
The FM-Index allows for the indexing of a whole string leading to much better memory requirements than similar previous tools which store all the k-mers explicitly in a sorted list. The indexed reference string can be one of the readily available reference genomes or a custom built string designed from multiple genomes (pangenomes). While the former option is undoubtedly easier to use, the latter method is interesting for further compression of the index itself.
The implementation of the FM-Index used in our program is the same as that of BWA. The indexed string is the hg38 human reference genome. The choice of relying on BWA for managing the index is justified by its wide availability among bioinformaticians and standard pipelines. It is very probable that a user of our program will have an already available index built with BWA that is compatible with the NGS data to compress.

Results

Here we present YALFF (Yet Another Lossy Fastq Filter), a tool for adjusting quality scores and reducing entropy leading to improved compressibility of FASTQ files. The performance of YALFF has been compared to other similar tools such as Leon and Quartz using a popular benchmark dataset (NA128736).

Leon relies on a de-Bruijn graph and a two-pass procedure on the input reads. In the first step it construct a probabilistic de-Bruijn graph from the input and then proceed to encode each read as a starting k-mer and a path inside the graph. The quality values are smoothed using the k-mer counting step during the first passage. On the other hand, Quartz is more similar to our algorithm because it only modify the quality scores and does not attempt to compress the genomic sequences at the same time. Both Quartz and YALFF work on FASTQ files either in input and in output without using a specialized file format like Leon.

The succinct dictionary allows YALFF to run on consumer computers with only 5.7 GB of available free RAM. As shown in Table 1 our smoothing algorithm outperforms those of the other tools leading to a better compression ratio while maintaining high accuracy and using less resources. Despite having a worse F-Measure than Quartz, YALFF does not degrade the precision of SNP calling, a desirable characteristic for medical applications.

Smoothing algorithm	Type	Precision	Recall	F-Measure	Time	RAM	Compression ratio
None (original files)	rf	0.9217	0.6341	0.7513	0.00	0.00	5.152
Leon	rf	0.9175	0.5582	0.6941	362	6.24	7.552
Quartz	rb	0.9180	0.6520	0.7624	2150	25.45	7.349
YALFF	rb	0.9216	0.6378	0.7539	2912	5.7	7.633

Table 1: Comparison of Precision, Recall, F-Measure in SNP calling and compression ratio between different tools. For each tool it is also reported its type (rf = reference free, rb = reference based), the real time measure in minutes using the `time` command as well as the peak memory consumption in GB. The compression ratio is defined as $\frac{\text{uncompressed size}}{\text{compressed size}}$ where the uncompressed size is 42608061544 bytes and the lossless compressor used is `bzip2`.