Referee: Reference Assembly Quality Scores

Gregg W.C. Thomas[1,2,]* and Matthew W. Hahn[1,2]

[1]Department of Biology, Indiana University, Bloomington

[2]Department of Computer Science, Indiana University, Bloomington

*Corresponding author: E-mail: grthomas@indiana.edu.

Accepted: April 17, 2019

Data deposition: No new sequence data were generated.

Abstract

Genome assemblies from next-generation sequencing technologies are now an integral part of biological research, but many sequencing and assembly processes are still error-prone. Unfortunately, these errors can propagate to downstream analyses and wreak havoc on results and conclusions. Although such errors are recognized when dealing with diploid genotype data, modern reference assemblies (which are represented as haploid sequences) lack any type of succinct quality assessment for every position. Here we present Referee, a program that uses diploid genotype quality information in order to annotate a haploid assembly with a quality score for every position. Referee aims to provide an assembly with concise quality information on a Phred-like scale in FASTQ format for easy filtering of low-quality sites. Referee also provides output of quality scores in BED format that can be easily visualized as tracks on most genome browsers. Referee is freely available at https://gwct.github.io/referee/.

Key words: genomics, bioinformatics, quality scores.

Introduction

Reference assemblies are haploid representations of the genome sequence of a species. Their use is ubiquitous in modern genetic and evolutionary research, especially in comparative genomics studies. Such studies range from questions about phylogenetic relationships to analyses searching for targets of adaptive natural selection. The conclusions of all analyses depend on the accuracy of the reference sequence; however, both genome assembly methods and the underlying sequencing technologies are error-prone (Hubisz et al. 2011). This inevitably leads to errors in downstream analyses and conclusions (e.g., Mallick et al. 2009; Schneider et al. 2009; Prosdocimi et al. 2012).

Many technologies provide a measure of base accuracy for every position in a sequencing read in the form of the quality score. This score represents the log-scaled value of the probability that the called base is incorrect. However, when assembling reads from genomes, transcriptomes, or other reduced-representation sequencing approaches (e.g., Baird et al. 2008) this quality information is lost. Here we present Referee, a program that provides a measure of the underlying quality for an assembled reference sequence. Referee uses genotype likelihoods, which are standard in resequencing studies (e.g., Li et al. 2008), to calculate a haploid reference quality score. The quality score, Q_R, ranges between 0 and 90 and represents the confidence we have that the called base at that position is correct. For positions where we have no confidence in the called base, Referee can suggest an alternate, better-scoring base. While tools do exist that examine assembly quality at a per-base level (Hunt et al. 2013), Referee aims to produce an easily interpretable quality score for any type of assembly, using any sequencing technology. These scores can then be used to inform any downstream analysis.

Materials and Methods

Referee uses the genotype likelihoods of all ten possible diploid genotypes at a site to calculate the quality score, Q_R, of the single base in the reference sequence. Referee summarizes the diploid genotype likelihoods for the haploid representation of the assembly by taking the sum of the likelihoods of the genotypes that contain the called base (L_{match}) and the sum of those that do not contain the called base (L_{mismatch}). For instance, if the called base is A, then $L_{\text{match}} = L(AA) + L(AT) + L(AC) + L(AG)$ and $L_{\text{mismatch}} = L(TT) + L(TC) + L(TG) + L(CC) + L(CG) + L(GG)$.

Taking the log-scaled ratio of these two sums gives us a quality score:

$$Q_R = \log\left(\frac{L_{\text{match}}}{L_{\text{mismatch}}}\right).$$

This scoring has the desirable behavior of being positive when we think the called reference base is correct and negative when we think it is incorrect due to lack of support; scores close to 0 indicate uncertainty in the called base. For sites that show more support for an alternate base call (i.e., sites with $Q_R \leq 0$), Referee can calculate Q_R for each of the three alternate bases and suggest the highest scoring base for that position.

Genotype Likelihoods

Referee's quality score requires genotype likelihoods from the reference individual. Such likelihoods are calculated by mapping the reads used in generating the assembly back to the reference assembly. Referee can calculate genotype likelihoods if given a pileup file as input. For this calculation we have implemented the Bayesian model of genotype likelihood developed by McKenna et al. (2010), with the additional consideration of mapping quality.

Referee also accepts genotype log-likelihoods as input from any method provided that they are formatted correctly. For example, the program ANGSD (Korneliussen et al. 2014) has the capability to output all ten genotype log-likelihoods in a format readily acceptable by Referee. Note that although ANGSD scales log-likelihoods by subtracting the highest score from all scores, this has no effect on Referee's calculations.

Referee's Scoring System

Because the quality score calculated by Referee is a ratio of probabilities, theoretically any score from negative to positive infinity is possible. In practice, scores tend to be limited to a range of −300 to +300 and have a strong correlation with read-depth (supplementary fig. S1, Supplementary Material online). For practical reasons, Referee's standard output limits the scores to a range of 0–90. This means that any negative score is converted to a score of 0, and any score above 90 is converted to 90. This makes the scores easily interpretable on a Phred-like scale and allows for conversion to ASCII characters for condensed FASTQ output.

Table 1
Referee Score Special Cases

Scenario	Q_R Score
$L_{\text{mismatch}} = 0$	91
Reference base called as *N*	−1
No reads mapped to site	−2

There are several scenarios in which it is not possible to calculate a quality score (table 1): In cases of very high read-depth, with all or most reads supporting the called base, it is possible that the sum of likelihoods for genotypes that do not contain the reference base (L_{mismatch}) will be 0. In these cases we are confident that the reference base is correct and assign a score of 91. If the reference base is an *N* or if no reads have mapped to the site we have no way of calculating Q_R, so we assign scores of −1 and −2, respectively, to indicate our uncertainty. In order to accommodate the −1 and −2 scores, quality scores are output as ASCII characters corresponding to $Q_R + 35$ (note that this scaling differs slightly from the standard Phred conversion).

Results

Referee is implemented entirely in Python, compatible with versions 2.7 and above and is freely available (https://gwct.github.io/referee/). Referee takes as input a single reference FASTA file representing the reference assembly and either pre-calculated genotype log-likelihoods or a pileup file from which it can calculate genotype likelihoods. Referee will output quality scores for every position in the input FASTA in either a simple tab delimited format (akin to the pileup) or in FASTQ format, with quality scores being converted to ASCII characters. Referee can also output quality scores in BED format, which can be used for visualizing tracks of scores in most genome browsers. Figure 1 shows a 100-kb stretch of Referee quality scores on chromosome 19 of the baboon genome (papAnu v2.0) in the UCSC Genome Browser (Kent et al. 2002).

Referee is intended for use on assemblies of any size, and from any technology that provides reads with base quality

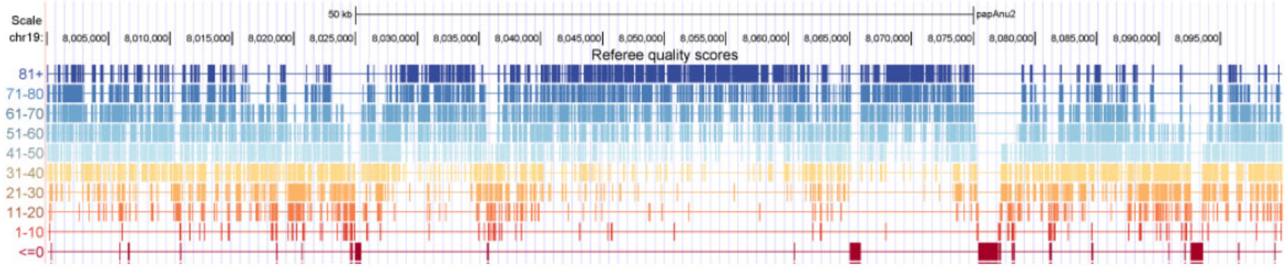

Fig. 1—Reference quality scores visualized for a 100,000 bp stretch of chromosome 19 in the baboon genome (*Papio anubis* v2.0) on the UCSC Genome Browser (http://genome.ucsc.edu).

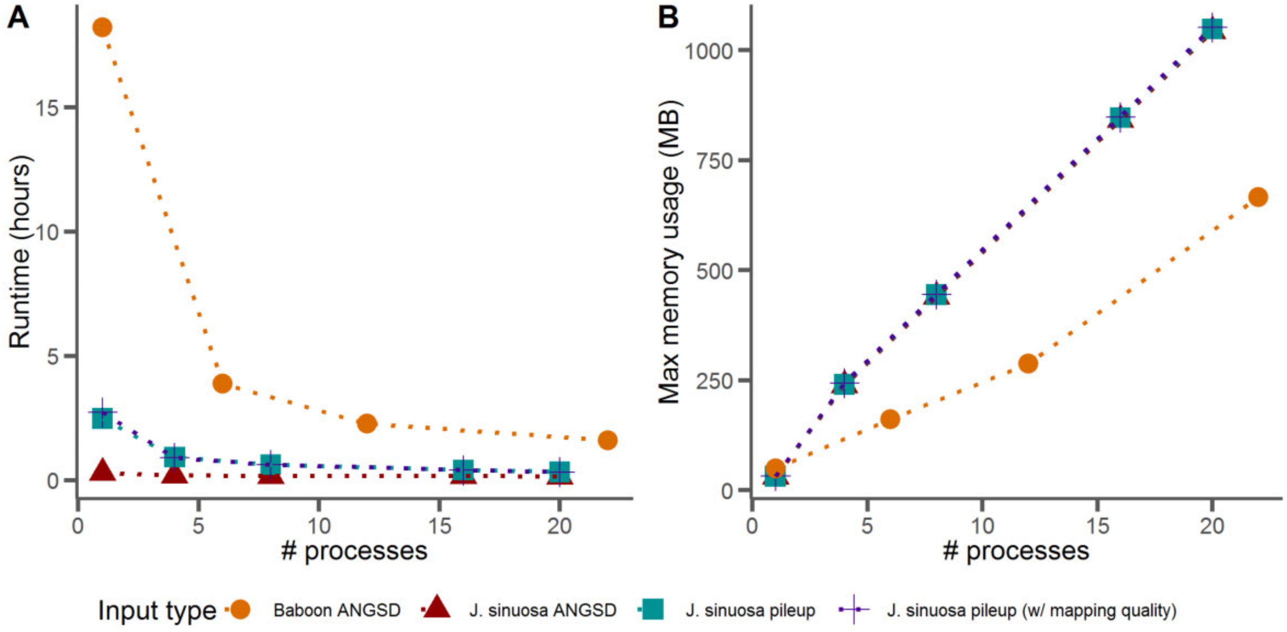

FIG. 2—Referee's run time (*A*) and max memory usage (*B*) on the *Jaltomata sinuosa* transcriptome and baboon genome. Note the memory improvement in the baboon genome data compared with the *J. sinuosa* transcriptome data as a result of splitting the input files by chromosome. ANGSD: genotype log-likelihoods were precalculated with the ANGSD software package (Korneliussen et al. 2014) and given as input to Referee; pileup: A pileup was created from the mapped reads which Referee used to calculate genotype likelihoods using only the base quality scores from the reads; pileup (w/mapping quality): The mapping qualities for the reads were included in the pileup and incorporated into Referee's genotype likelihood calculations.

scores (e.g., Illumina or Oxford Nanopore). To make it scalable with even the largest of today's sequenced genomes, Referee is designed to use multiple processes without a large memory footprint. We tested the performance of Referee on two data sets: a transcriptome assembly from *Jaltomata sinuosa* (Wu et al. 2018) using Illumina RNA-seq reads (SRA accession SRX2676125) and a genome assembly from the baboon, *Papio anubis* (GCF_000264685.2) using only the Illumina paired-end reads that were used in the assembly process (SRA accessions: SRR927653, SRR927654, SRR927655, SRR927656, SRR927657, SRR927658, SRR927659). Test runs were done on Indiana University's Carbonate computer cluster (Red Hat Enterprise 7.x with 256 GB of RAM and two 12-core Intel Xeon E5-2680 v3 CPUs). For *J. sinuosa* the reads were assembled with Trinity (Grabherr et al. 2011) and for both species reads were mapped back to their respective assemblies with BWA (Li and Durbin 2009). We find that for the *J. sinuosa* transcriptome, even when utilizing only one process, Referee completes in 20 min with precalculated genotype likelihoods. Unsurprisingly, calculating the likelihoods is detrimental to run time, raising it to 2.73 h, but allocating additional processes more than makes up for this time loss (fig. 2*A*). For the much larger baboon genome data set we observe a run time of 18 h when using precalculated genotype likelihoods. Again this is drastically reduced to 1.6 h when using multiple processes (fig. 2*A*). Memory usage never exceeds 1 GB (fig. 2*B*). This makes Referee widely usable regardless of operating system.

Conclusions

The wide-ranging applicability of genome assemblies in modern biological research means their accuracy is of utmost importance in order to reach unambiguous conclusions. Evolutionary inferences into species relationships and the targets of positive selection depend on this accuracy. Referee adds a simple step between the assembly and analysis of a genome to improve the assembly for all purposes. By accounting for the underlying base quality in the reads and the diploid nature of most genome assemblies, Referee's scores can be used to inform researchers of sites to filter from their analyses or of better scoring alternate bases. This is accomplished through a fast and easy to use software package: https://gwct.github.io/referee/.

Supplementary Material

Supplementary data are available at *Genome Biology and Evolution* online.

Acknowledgment

We would like to thank Haixu Tang for his guidance. This work was funded by the Precision Health Initiative at Indiana University.

Literature Cited

Baird NA, et al. 2008. Rapid SNP discovery and genetic mapping using sequenced RAD markers. PLoS One 3(10):e3376.

Grabherr MG, et al. 2011. Full-length transcriptome assembly from RNA-seq data without a reference genome. Nat Biotechnol. 29(7):644–652.

Hubisz MJ, Lin MF, Kellis M, Siepel A. 2011. Error and error mitigation in low-coverage genome assemblies. PLoS One 6(2):e17034.

Hunt M, et al. 2013. REAPR: a universal tool for genome assembly evaluation. Genome Biol. 14(5):R47.

Kent WJ, et al. 2002. The human genome browser at UCSC. Genome Res. 12(6):996–1006.

Korneliussen TS, Albrechtsen A, Nielsen R. 2014. ANGSD: analysis of next generation sequencing data. BMC Bioinformatics 15:356.

Li H, Durbin R. 2009. Fast and accurate short read alignment with Burrows–Wheeler transform. Bioinformatics 25(14):1754–1760.

Li H, Ruan J, Durbin R. 2008. Mapping short DNA sequencing reads and calling variants using mapping quality scores. Genome Res. 18(11):1851–1858.

Mallick S, Gnerre S, Muller P, Reich D. 2009. The difficulty of avoiding false positives in genome scans for natural selection. Genome Res. 19(5):922–933.

McKenna A, et al. 2010. The genome analysis toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. Genome Res. 20(9):1297–1303.

Prosdocimi F, Linard B, Pontarotti P, Poch O, Thompson JD. 2012. Controversies in modern evolutionary biology: the imperative for error detection and quality control. BMC Genomics 13:5.

Schneider A, et al. 2009. Estimates of positive Darwinian selection are inflated by errors in sequencing, annotation, and alignment. Genome Biol Evol. 1:114–118.

Wu M, Kostyun JL, Hahn MW, Moyle LC. 2018. Dissecting the basis of novel trait evolution in a radiation with widespread phylogenetic discordance. Mol Ecol. 27(16):3301–3316.

Associate editor: Belinda Chang