BIOINFORMATICS ORIGINAL PAPER

Vol. 29 no. 4 2013, pages 435–443
doi:10.1093/bioinformatics/bts723

Genome analysis

Advance Access publication January 9, 2013

ALE: a generic assembly likelihood evaluation framework for assessing the accuracy of genome and metagenome assemblies

Scott C. Clark[1], Rob Egan[2,3], Peter I. Frazier[4,*] and Zhong Wang[2,3,*]

[1]Center for Applied Mathematics, Cornell University, Ithaca, NY 14853, USA, [2]Department of Energy, Joint Genome Institute, Walnut Creek, CA 94598, USA, [3]Genomics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA and [4]School of Operations Research and Information Engineering, Cornell University, Ithaca, NY 14853, USA

Associate Editor: Alex Bateman

ABSTRACT

Motivation: Researchers need general purpose methods for objectively evaluating the accuracy of single and metagenome assemblies and for automatically detecting any errors they may contain. Current methods do not fully meet this need because they require a reference, only consider one of the many aspects of assembly quality or lack statistical justification, and none are designed to evaluate metagenome assemblies.

Results: In this article, we present an Assembly Likelihood Evaluation (ALE) framework that overcomes these limitations, systematically evaluating the accuracy of an assembly in a reference-independent manner using rigorous statistical methods. This framework is comprehensive, and integrates read quality, mate pair orientation and insert length (for paired-end reads), sequencing coverage, read alignment and k-mer frequency. ALE pinpoints synthetic errors in both single and metagenomic assemblies, including single-base errors, insertions/deletions, genome rearrangements and chimeric assemblies presented in metagenomes. At the genome level with real-world data, ALE identifies three large misassemblies from the *Spirochaeta smaragdinae* finished genome, which were all independently validated by Pacific Biosciences sequencing. At the single-base level with Illumina data, ALE recovers 215 of 222 (97%) single nucleotide variants in a training set from a GC-rich *Rhodobacter sphaeroides* genome. Using real Pacific Biosciences data, ALE identifies 12 of 12 synthetic errors in a Lambda Phage genome, surpassing even Pacific Biosciences' own variant caller, EviCons. In summary, the ALE framework provides a comprehensive, reference-independent and statistically rigorous measure of single genome and metagenome assembly accuracy, which can be used to identify misassemblies or to optimize the assembly process.

Availability: ALE is released as open source software under the UoI/NCSA license at http://www.alescore.org. It is implemented in C and Python.

Contact: pf98@cornell.edu or ZhongWang@lbl.gov

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on June 22, 2012; revised on December 18, 2012; accepted on December 21, 2012

*To whom correspondence should be addressed.

1 INTRODUCTION

Recent advances in next-generation, high-throughput sequencing technologies have dramatically reduced the cost of sequencing (Metzker, 2010). With the development of genome assemblers able to use large volumes of sequence data, reference genomes are now rapidly produced using the whole genome shotgun strategy, from small, simple microbial genomes (Wu *et al.*, 2009) to large, complex plant or mammalian genomes (Fujimoto *et al.*, 2010; Li *et al.*, 2010; Schmutz *et al.*, 2010; Zimin *et al.*, 2008). Meanwhile, genomes are also being generated directly from complex communities using culture-independent approaches, including singe-cell genome sequencing and metagenome sequencing (Hess *et al.*, 2011; Iverson *et al.*, 2012; Woyke *et al.*, 2010; Yilmaz *et al.*, 2011). The ability to assemble a metagenome is particularly important because resolving the genomes of individual species, or at least the most abundant, from a complex community is crucial to exploring inter-species interactions and understanding the community's structure, dynamics and function.

Assembly of individual genomes from NGS datasets poses significant informatics challenges, including short read length, noisy data and large data volume (Lin *et al.*, 2011; Pop, 2009). Owing to these challenges, errors widely exist in single genome assemblies derived from NGS datasets, with different specific errors commonly associated with particular datasets, genomes and tools (Haiminen *et al.*, 2011). Beyond the challenges faced in assembling single genomes, metagenome assembly poses unique additional challenges. First, although the sequence depth of a single genome should be approximately uniform, the sequence depth of genomes in a metagenome varies greatly. Second, the difficulty of resolving repetitive regions within a single genome is exacerbated in metagenome assembly because conserved genomic regions and lateral gene transfer greatly increase the portion of falsely identified repetitive genomic regions. Despite these unique challenges, assemblers designed for single genomes are being applied to metagenome data without being significantly modified to systematically address errors introduced in this way (Hess *et al.*, 2011; Iverson *et al.*, 2012; Qin *et al.*, 2010).

Several tools have been developed to detect errors in single genome assemblies. If a reference genome for the targeted organism is available, or one is available from a closely related species, erroneous insertions, deletions or large gaps can be detected by

comparative analysis of the reference and the genome assembly in question (Darling *et al.*, 2011; Earl *et al.*, 2011; Meader, 2010; Salzberg *et al.*, 2012). If a reference is unavailable, the alignment of the raw reads with their assembly provides indirect measures of assembly quality such as coverage depth and mate pair consistency. This information can then be used to detect single-base changes, repeat condensation or expansion, false segmental duplications and other misassemblies (Choi *et al.*, 2008; Narzisi and Mishra, 2011; Phillippy *et al.*, 2008; Vezzi *et al.*, 2012; Zimin *et al.*, 2008). Despite this progress, researchers still lack a method that integrates indirect measures of read alignment quality in a quantitative, comprehensive and statistically well-founded manner to systematically detect errors in single genome assemblies. Moreover, metrics suitable for evaluating metagenome assembly accuracy, and associated quantitative methods for detecting errors in metagenome assemblies, have yet to be developed.

In this work, we develop a novel statistical model for evaluating assembly accuracy in a reference-independent manner. Using Bayesian statistics, we give an expression for the probability that an assembly is correct, and provide an automated software tool Assembly Likelihood Evaluation (ALE) based on this expression. The provided tool may be used in three ways. First, it allows examining the contribution to this probability of correctness from each base in the assembly, which can be used to identify specific errors and their locations. This is particularly useful for genome finishing. Second, it provides an overall score for different assemblies of the same genome or metagenome, thereby enabling comparison of these assemblies and optimization of the assembly process. Third, when applying re-sequencing data to a reference genome, ALE can detect structural variations.

2 METHODS

2.1 The ALE score and the likelihood of an assembly

The ALE framework is founded on a statistical model that describes two probabilities: a Bayesian prior probability distribution $P(S)$ describing the likelihood of an assembly S without any read information and a probability $P(R|S)$ describing the likelihood of a set of reads, R, being generated from an assembly, S. The prior $P(S)$ can be computed using the k-mer distribution of the assembly, whereas the likelihood $P(R|S)$ is calculated from information about read quality, agreement between the mapped reads and the proposed assembly, mate pair orientation, insert length (paired-end reads) and sequencing depth. A detailed description of the likelihood and prior probability is given in the following.

The ALE score, except for a proportionality constant that depends on the reads but not on the assembly, is the logarithm of the probability that the assembly is correct, $P(S|R)$. According to Bayes' rule, this probability is

$$P(S|R) = P(R|S)P(S)/Z. \quad (1)$$

where Z is a proportionality constant ensuring $P(S|R)$ is a probability distribution. As is typical in large-scale applications of Bayesian statistics, computing Z exactly is intractable. The ALE score is computed by replacing Z with an approximation described in the Supplementary Materials, and then taking the logarithm of the resulting approximation to $P(S|R)$.

The ALE score can be used to compare two different assemblies of the same genome, S_1 and S_2. Call A_1, the ALE score of the first assembly, and A_2, the ALE score of the second, both generated from the same set of reads R. The difference of these scores is then given by the equation

$$A_1 - A_2 = \log\left(\frac{P(S_1|R)}{P(S_2|R)}\right). \quad (2)$$

The assembly with the higher ALE score is also the one with the larger probability of being correct. Moreover, the difference between two assemblies' ALE scores describes their relative probabilities of correctness. Below, we refer to the ALE score more precisely as the total ALE score, to differentiate it from the sub-scores (described later in the text) used to construct it.

Although the ALE score can be reported as a standalone value, this is made possible only to facilitate comparisons with other assemblies of the same genome. We emphasize that the ALE score is a comparative measure and should not be used to judge the quality of a single assembly *in isolation*, as errors in estimating Z may cause a large difference between the ALE score and $\log(P(S|R))$. We also emphasize that the ALE score should only be used to compare different assemblies of the same genome, for which the ALE scores have been calculated using the same set of reads.

Figure 1 shows the pipeline used to compute the total ALE score. Given a set of reads and a proposed assembly, ALE first takes as input the alignments of the reads onto the assembly in the form of a SAM or BAM file (Li *et al.*, 2009), which can be produced by a third-party alignment algorithm such as bowtie (Langmead *et al.*, 2009) or bwa (Li *et al.*, 2009). ALE then determines the probabilistic placement of each read and a corresponding placement sub-score for each mapped base, which describes how well the read agrees with the assembly. In the case of paired-end reads, ALE also calculates an insert sub-score for all mapped bases of the assembly from the read pair, which describes how well the distance between the mapped reads matches the distribution of lengths that we would expect from the sequencing library. This insert sub-score is similar to the compression-expansion (CE) statistic of Zimin *et al.* (2008) with details given in the Supplementary Materials. ALE also calculates a depth sub-score, which measures the evenness of the sequencing depth accounting for the GC bias prevalent in some NGS techniques. The placement, insert and depth sub-scores together determine $P(R|S)$. Independently, with only the assembly and not the reads, ALE calculates the k-mer sub-score and the prior $P(S)$. Each sub-score is calculated for each scaffold or contig within an assembly independently, allowing for genome variations commonly found in metagenomes because each contig/scaffold is likely from a different species with a different k-mer profile. The four sub-scores are then combined to form the total ALE score. The constituent calculations in this pipeline are described in the Supplementary Material.

The contributions to these four sub-scores are reported by ALE as a function of position within the assembly and can be visualized with the included plotting package or exported to genome viewers including the Integrative Genomics Viewer (Nicol *et al.*, 2009) and the UCSC genome browser (Kent *et al.*, 2002).

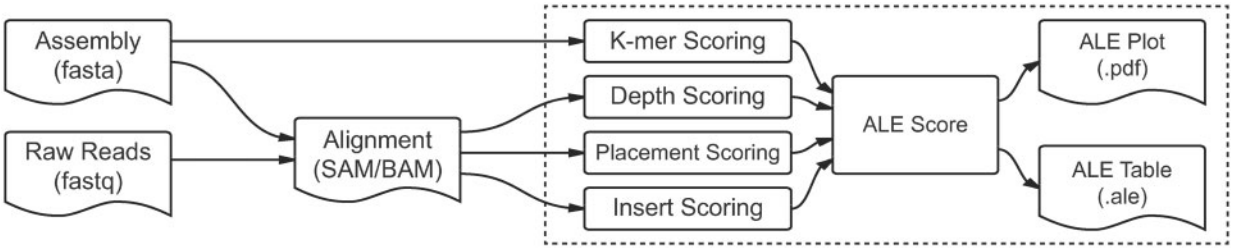

Fig. 1. The components of the total ALE score. ALE takes a proposed assembly and an alignment of reads as input. Four scores, the k-mer, placement, depth and insert sub-scores are computed using the model described in Section 2. From the four scores, a total ALE score is calculated and reported as a text file (.ale), and the text file can be used for input into the supplied plotter to generate a PDF file for visualization

2.2 Details of the probabilistic ingredients of the ALE score

We now describe the four sub-scores (placement, insert, depth and k-mer) and the role they play within the ALE framework.

The first three of these sub-scores appear in the likelihood $P(R|S)$. ALE computes $P(R|S)$ using a probabilistic model for the way in which reads are generated from an assembly during the whole genome shotgun sequencing process. This model makes independence assumptions that decomposes this probability into a product of three terms,

$$P(R|S) = P_{\text{placement}}(R|S)P_{\text{insert}}(R|S)P_{\text{depth}}(R|S). \quad (3)$$

Each term is a separate sub-score and is explained later in the text in detail.

2.2.1 Placement $P_{\text{placement}}(R|S)$ quantifies how well the sequence of the reads agrees with the assembly. Assuming that every paired read r_i is generated independently from the assembly, the probability of a set of reads R given an assembly S is $P_{\text{placement}}(R|S) = \prod_{r_i \in R} P_{\text{placement}}(r_i|S)$, where $P_{\text{placement}}(r_i|S)$ is itself the product of two independent probability distributions, $P_{\text{placement}}(r_i|S) = P_{\text{matches}}(r_i|S)P_{\text{orientation}}(r_i|S)$.

Here, $P_{\text{matches}}(r_i|S)$ describes how well the read matches the subsection of the assembly to which it maps, and $P_{\text{orientation}}(r_i|S)$ describes whether the mate pairs have an orientation that is consistent with the library. We now describe in detail how these two probabilities are computed, beginning with $P_{\text{matches}}(r_i|S)$.

Assuming that each base j of the read is correctly called by the sequencer independently with a probability equal to the base's quality score Q_j, we have $P_{\text{matches}}(r_i|S) = \prod_{\text{base}_j \in r_i} P(\text{base}_j|S)$, where $P(\text{base}_j|S) = Q_j$ when the base j correctly matches the assembly and $P(\text{base}_j|S) = (1 - Q_j)/4$ when it does not. This expression follows from our modeling assumption that all four possible errors that the sequencer could have reported (three different substitutions and deletion) are equally likely when the read does not match the sequence. If the assembly has an unknown base (denoted 'N'), we set $P(\text{base}_j|S) = 1/4$, modeling the lack of information about the correct base at that location. If an ambiguity code is reported by the sequencer, then the aforementioned expression is modified to account for the distribution over the possible bases encoded by that code. Each read may only be 'placed' at a single position in the assembly. If the aligner placed a particular read at more than one position, we choose one position at random, weighting by $P_{\text{placement}}(r_j|S)$. This allows for repeat regions to be properly represented with the correct number of reads in expectation.

The orientation likelihood, $P_{\text{orientation}}(r_i|S)$, is calculated by first counting the number of times that each orientation occurs in each library from the mapping information. The likelihood $P_{\text{orientation}}(r_i|S)$ is then the empirical frequency of the observed orientation of the read r_i in the library to which r_i belongs. (This likelihood can also be overridden with user-specified values.)

We also derive per-base placement sub-scores at each position in the assembly. The placement sub-score at a particular position is the geometric mean $\left[\prod_{r_i} P_{\text{placement}}(r_i|S)\right]^{1/N}$, where the product is over all reads r_i covering the given position, and N is the number of such reads.

2.2.2 Insert $P_{\text{insert}}(R|S)$ describes how well the mate pairs' insert lengths match those we would expect and is computed as $P_{\text{insert}}(R|S) = \prod_{r_i \in R} P_{\text{insert}}(r_i|S)$. The insert likelihood, $P_{\text{insert}}(r_i|S)$, is determined by first observing all insert lengths from all mappings of all reads and calculating the population mean, μ, and variance, σ^2, of these lengths (the mean and variance can also be set by the user, if they are known). This step only needs to be done once. Once completed, we calculate the insert likelihood for each read pair r_i by assuming that the observed insert length L_i is distributed normally with this mean and variance, $P_{\text{insert}}(r_i|S) = \text{Normal}(L_i; \mu, \sigma^2)$

As with the placement sub-score, we calculate the insert sub-score at a position as the geometric mean of the $P_{\text{insert}}(r_i|S)$ of all reads r_i covering that position. This can identify areas of constriction or expansion within a proposed assembly.

The insert sub-score is similar to the CE statistics of Zimin *et al.* (2008), as we now show. To describe the similarity, we first write the insert sub-score as $\left[\prod_i (2\pi\sigma^2)^{-1/2} \exp\left(-\frac{(L_i-\mu)^2}{2\sigma^2}\right)\right]^{1/N} = (2\pi\sigma^2)^{-1/2} \exp\left(-\frac{1}{2\sigma^2 N}\sum_{i=1}^{N}(L_i-\mu)^2\right)$,

where the products and sums over i are over all reads covering a given position, and N is the number of such reads. We now use the fact that $\frac{1}{N}\sum_{i=1}^{N}(L_i-\mu)^2 = S^2 + (M-\mu)^2$, where $M = \frac{1}{N}\sum_i L_i$ is the sample mean of the implied insert lengths and $S^2 = \frac{1}{N}\sum_i (L_i - M)^2$ is the sample variance. The CE statistic from Zimin *et al.* (2008) is $\text{CE} = \frac{M-\mu}{\sigma/\sqrt{N}}$, which implies

$\frac{1}{\sigma^2 N}\sum_{i=1}^{N}(L_i-\mu)^2 = \frac{S^2}{\sigma^2} + N \times \mathrm{CE}^2$. As N grows large, the CE statistic is asymptotically normal, owing to the central limit theorem.

The log of the insert sub-score can then be written $-\frac{1}{2}[\log(2\pi\sigma^2) + (S^2/\sigma^2) + N \times \mathrm{CE}^2]$. The insert sub-score is decreasing in $\frac{S^2}{\sigma^2} + N \times \mathrm{CE}^2$, and the term involving the CE statistic dominates when N is large. Thus, when N is large, the positions with lowest insert sub-score are those positions whose CE statistic is furthest from 0. This, flagging those regions with low insert-sub score is similar to the rule recommended in Zimin *et al.* (2008) of flagging those regions with |CE| larger than a fixed cutoff value.

2.2.3 *Depth* $P_{\mathrm{depth}}(R|S)$ describes how well the depth at each location agrees with the depth that we would expect, given the GC content at that location. It is the product of a depth sub-score over all positions in the assembly, $P_{\mathrm{depth}}(R|S) = \prod_i P_{\mathrm{depth}}(d_i|S)$, where d_i is the depth at position i.

The depth d_i is ideally Poisson-distributed (Lander and Waterman, 1988). However, most next-generation sequencers and library preparation techniques can bias GC-rich areas of a genome (Aird *et al.*, 2011). This bias affects the observed depth in specific areas. We model the depths as Poisson distributed about a mean drawn from an independent Gamma distribution centered at the expected depth for that position, given its GC content. This models our uncertainty about the mean of the Poisson distribution, arising from the dependence of the expected depth on more than just the GC content at that position, including 'hard stops', and the GC content at nearby positions. It results in an infinite mixture of Poissons that is equivalent to a Negative Binomial distribution.

We first calculate for each of the following 100 ranges of GC content, 0–1, 1–2, …, 99–100%, the average observed depth over positions in the assembly whose GC content (calculated as the GC content within an average read length) is within this range. Let $\mu_{\mathrm{depth}(X_i)}$ be the average observed depth for the GC content range in which X_i falls, where X_i is the GC content percentage averaged across all reads that map (in the placement step) to that position. If any $\mu_{\mathrm{depth}}(X_i)$ falls below a minimum value of 10, we use this minimum value instead. This discounts regions of exceptionally low average depth. Then, at any given position i, the depth sub-score is $P_{\mathrm{depth}}(d_j|S, X_i) = \int_0^\infty \mathrm{Poisson}(d_j; Y_i)\mathrm{Gamma}(Y_i; \max(10, \mu_{\mathrm{depth}(X_i)}), 1)\, dY_i = \mathrm{NegBinom}(d_j; \max(10, \mu_{\mathrm{depth}(X_i)}), 1/2)$.

2.2.4 *k-mer* $P_{\mathrm{kmer}}(S) \propto P(S)$ describes the likelihood of the assembly S, in the absence of any read information. Within this prior probability distribution, we encode the belief that within a single genome, each k-mer (a permutation of k base pairs, where k is a fixed user defined number initially set to 4) has a unique k-mer frequency. The 4^k dimensional vector giving this frequency for each k-mer is conserved across a genome and can help determine if two different genomes have been mistakenly combined (Teeling *et al.*, 2004; Woyke *et al.*, 2006). Let K be the set of all possible unique k-mers, so $|K| = 4^k$, and for each i in K let n_i be the number of times this k-mer appears in a contig in the assembly. Then, the frequency f_i of a particular k-mer i within a contig is $f_i = n_i / \sum_{j \in K} n_j$. The k-mer score is the product of this frequency over each k-mer appearing in each contig of the assembly S, which can be written as $P_{kmer}(S) = \prod_{i \in K} f_i^{n_i}$. This is equivalent to assuming each k-mer in the assembly is drawn independently from a common multinomial distribution with probabilities empirically estimated from the assembly.

This prior distribution does not account for horizontal gene transfer, e.g. from phages, and thus may inappropriately flag such regions as being misassembled.

The k-mer sub-score of a base at any given position in the assembly is the geometric average of $P_{\mathrm{kmer}}(S)$ of all k-mers that cover that position. In calculating this average, the very first base in the genome only has one contributing k-mer, the second has two, up to k contributing k-mers after $k-1$ bases.

3 RESULTS

3.1 Performance on major types of misassemblies in a genome assembly with synthetic data

Common assembly errors include single-base substitutions, insertion/deletions, chimeric assemblies derived from translocations or misjoins and copy number errors derived from repeat condensation/expansions. To test ALE's ability to detect such errors, we generated synthetic reads from a reference genome and then seeded the reference with each type of error. First, 400 000 pair-end synthetic reads were generated from the first 350 Kb at random positions of *Escherichia coli K12* Substrain DH10B (Durfee *et al.*, 2008), with insert length normally distributed with mean 200 b and standard deviation 7 b. Next, synthetic misassemblies were introduced at six locations within this reference. The misassemblies introduced were a substitution, insertion, deletion, inversion, translocation and a copy number error, respectively. We treated this mutated genome as the proposed assembly.

We tested ALE by aligning the aforementioned synthetic reads to the proposed assembly using bowtie (Langmead *et al.*, 2009) and ran the results through the ALE software package. The ALE plotter automatically thresholds each error and produces plots of the sub-scores near each error (see the Supplementary Materials). We found that ALE is able to locate each type of error in the proposed assembly. At the genome level, as shown in Figure 2, at least one of the four sub-scores drops dramatically in each region containing a synthetic error and reports no false discoveries. These results suggest that ALE systematically reports all major types of errors with simulated data.

Furthermore, the total ALE score decreases, as more errors are added to the assembly. As shown in Figure 3A, as the number of substitution, insertion and deletion errors increases, the total ALE score decreases monotonically, at a rate determined by the quality scores of the data (see Section 2). This suggests that the total ALE score indicates overall assembly accuracy.

3.2 Detecting chimeric assemblies in a synthetic metagenome

One common assembly error in metagenome assemblies is a chimeric assembly consisting of two or more genomes. To test ALE's ability to distinguish this type of metagenome-specific error, we simulated a misassembled contig by joining several pieces of two genomes in random order. This use of a

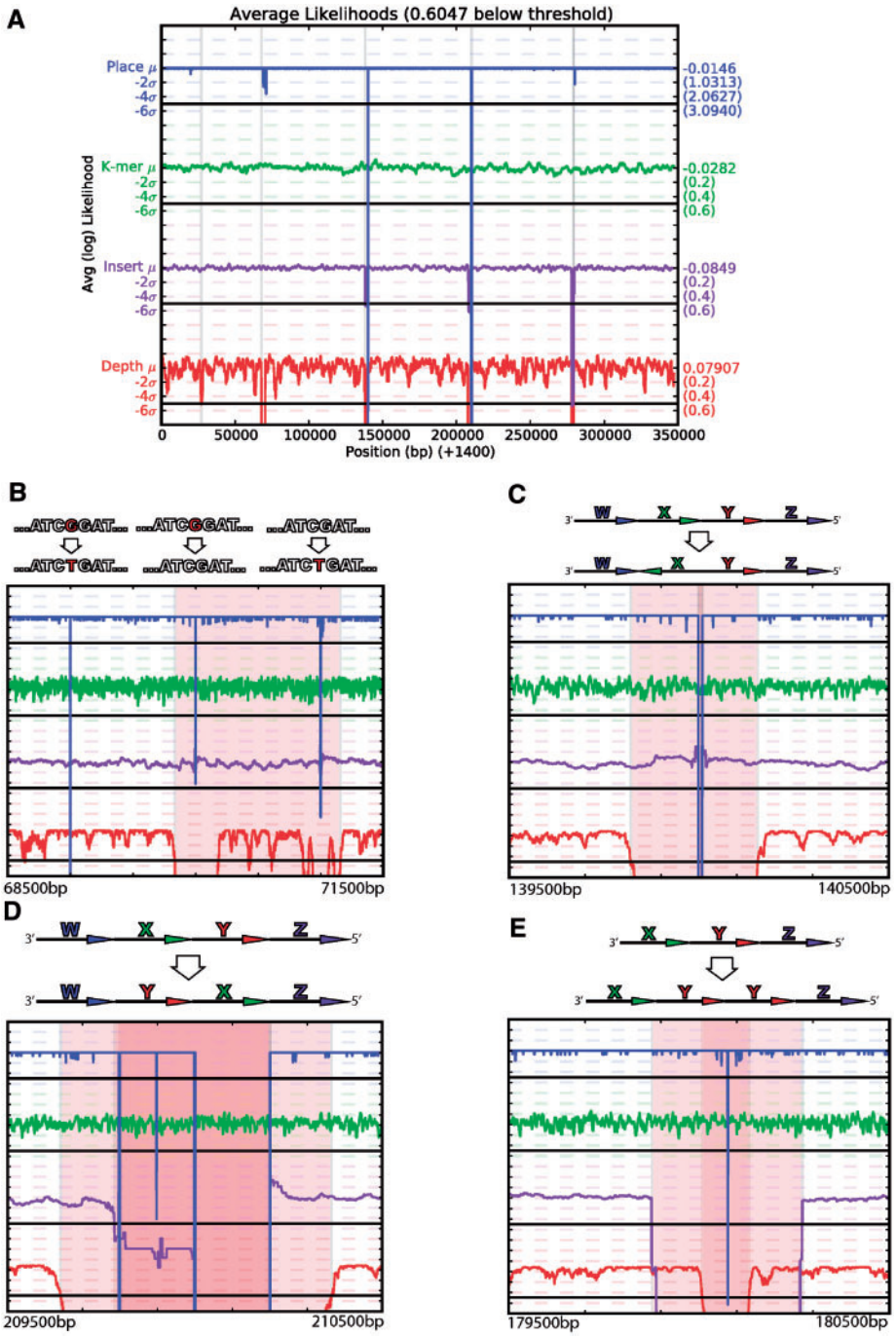

Fig. 2. The performance of ALE on synthetic errors in *E. coli*. At the genome level, at least one of the four sub-scores drops dramatically in each region containing a synthetic error (**A**). A higher resolution view for each type is illustrated in (**B–E**). (B) Single-base substitution, deletion and insertion errors; (C) an inversion error of length 200 b; (D) a transposition error of length 200 b; and (E) a copy number error of length 77 b. Diagrams above each plot illustrate how each error was generated. Implementation details are within the image_maker.py script in the ALE code base

known, synthetic reference is required for testing ALE's sensitivity to chimeric metagenomes in an unbiased way, as there are no true metagenome references available. In these tests, the k-mer score drops markedly at the interface of the two genomes in the synthetic metagenome, as shown in the Supplementary Fig. S1.

ALE relies on the k-mer sub-score (the default is $k = 4$) to distinguish contigs coming from different microbial species, as tetra-nucleotide frequencies are a reliable species-specific signature (Teeling *et al.*, 2004; Woyke *et al.*, 2006). If a genome, or contig, contains two or more distinct regions characterized by different k-mer vectors, then the k-mer sub-score will be lower

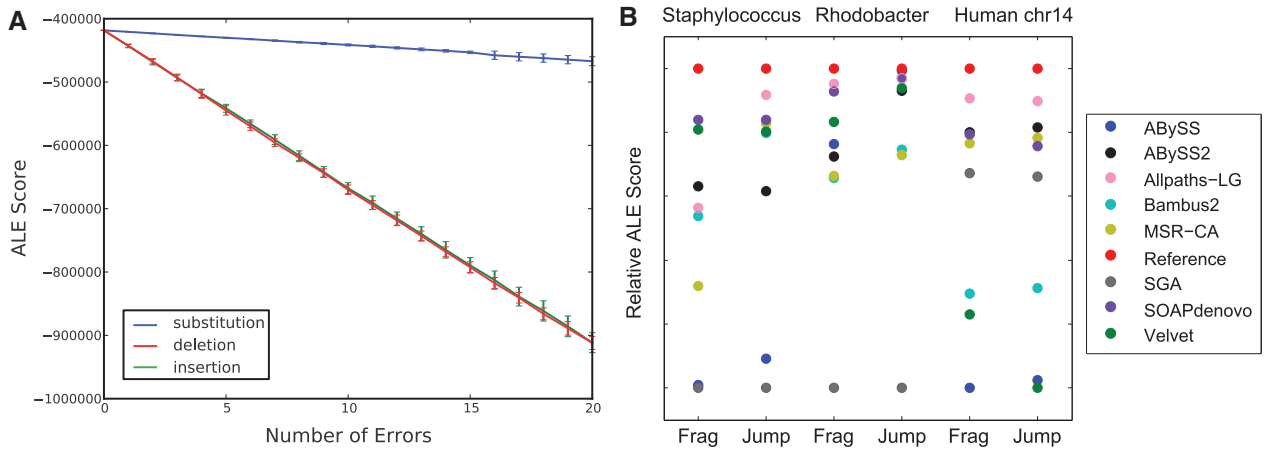

Fig. 3. ALE scores indicate overall assembly accuracy. (**A**) ALE scores monotonically decrease as the number of synthetic errors increase in an *E. coli* genome assembly. (**B**) ALE scores correlate with the accuracy of assemblies of three genomes with two independent sequencing libraries. Frag: Short fragment library. Jump: short jumping library. For each library, ALE scores from each assembly were scaled to [0, 1] to get relative ALE scores

for the positions characterized by the less prevalent k-mer vector (see Section 2). Because the other sub-scores are unaffected by the mixture, the drop in total ALE score is owing to the lower k-mer sub-score. This unique capability of ALE allows easy detection of chimeric contig/scaffolds within a metagenome assembly.

3.3 Discovery of real errors in a genome assembly using real data

The aforementioned experiments used simulated reads or assemblies with simulated errors. Noise in real reads, and real errors in genome/metagenome assemblies, often has a complex structure, presenting an additional challenge to ALE. To test ALE using real world assemblies with real reads, we chose a finished genome, *Spirochaeta smargdinae* DSM 11293, originally constructed from 454 and Illumina reads (Mavromatis *et al.*, 2010) and applied ALE to it using one lane of 2×76 paired-end Illumina reads. The results are shown in the Supplementary Fig. S2. At the genome level, ALE found five errors, including a large 560 Kb region (3.91–4.48 Mb) in the proposed assembly where the depth sub-score dropped below the threshold. We found three areas producing errors that are likely due to repeat condensation. For example, further examination of two regions (408–415 Kb and 4.241–4.247 Mb) by overlaying the Illumina short read data indicates these regions have much higher sequence depth (2×) than neighboring regions and contain many SNPs (two alleles of roughly equal ratio) (Supplementary Fig. S2B and C), supporting the hypothesis that there are two copies of these regions in this genome. The boundaries of these regions also have abnormal placement and insert sub-scores, further supporting the hypothesis that there are misassemblies at the aforementioned locations.

To determine whether these errors identified by ALE are true assembly errors or Illumina sequencing artifacts, we independently validated the results using Pacific Biosciences (PacBio) sequencing data. A total of 53 SMRT cells comprising 221 Mb of mapped reads or 34 folds of coverage were aligned to the assembly. Manual inspection of the resulting PacBio alignment confirms 5/5 assembly errors (Supplementary Fig. S2B and C), suggesting the errors identified by ALE are true errors in the assembly. The locations of these errors, and sub-score that caused the violation, are given in the Supplementary Table S1.

3.4 Total ALE scores and genome assembly accuracy

Assemblies generated by eight different assembly protocols from three datasets (*Staphylococcus aureus*, *Rhodobacter sphaeroides* and Human chromosome 14) were selected from the GAGE study (Salzberg *et al.*, 2012) to evaluate how well ALE scores reflect assembly accuracy. As a positive control, the reference genome was evaluated in parallel with the other assemblies. For all three genomes, Illumina sequencing data from both a short fragment and a short jumping library were used to generate ALE scores. Total ALE scores were generated for each assembly and each library as described previously. Results are shown in Figure 3B.

Although that there is no quantitative measurement of assembly accuracy (other than ALE itself) for use in validation, the GAGE study did indicate the AllPaths-LG and SOAPdenovo usually give more accurate assemblies. In addition, the reference genome should represent an assembly with best quality. As shown in Figure 3B, ALE scores are consistent with these notions. Among all the assemblies, ALE scores consistently indicate the reference genomes are the best assemblies, and assemblies from AllPaths-LG and SOAPdenovo consistently have better ALE scores than those from other assemblers.

3.5 Sensitivity to single nucleotide variations in real data

We tested ALE's ability to detect single base errors in real data from a resequencing project. In this project, one lane of Illumina 36×2 paired reads was generated from a new strain of *R. sphaeroides* 2.4.1 (Choudhary *et al.*, 2006) with an insert length of 200 b covering the genome with an average coverage depth of 557. This genome has a high GC content (68%) and contains 336 hard stops and many more low depth regions. A hard stop is a region

where a bias in the sequencer causes it to report 0 depth (no reads) without any read pairs spanning the region. Because low-coverage regions make single nucleotide variation (SNV) detection less reliable for many SNP detectors (Wang *et al*., 2011), we excluded such regions and manually compiled a reference set of 222 possible SNVs for this strain (176 from Chromosome1, length 3.2 Mb; 46 from Chromosome2, length 0.94 Mb). The placement sub-score was then computed using the aligned reads.

To enumerate the positions that ALE found to detract the most from the assembly's probability of correctness, we sorted the placement sub-scores for each chromosome. The 0.0001% worst scoring positions (219 regions) on Chromosome1 are within a read length of 154 of the 176 variants (88%), and the top 0.0005% worst positions (977 regions) are within a read length of >97% of the variants. The same experiment for Chromosome2 recovers 87% (40 of 46 variants from 63 regions) and 96% (from 309 regions), respectively. This shows that the positions at which the proposed assembly differs from the genome generating the reads are among the positions with the worst sub-scores. The regions with poor sub-scores that do not correspond to the variant list are other regions of the assembly unsupported by the read evidence, such as hard stops regions of low coverage that stem from the bias of the sequencer. This shows that ALE can locate regions unsupported by the read evidence, including SNVs, and that ALE accurately gauges assembly accuracy at single base resolution.

3.6 ALE's sensitivity and specificity

Six assemblies of *S. aureus* were selected (Salzberg *et al*., 2012) to evaluate ALE's ability to detect assembly errors Figure 4, leaving out an assembly from SGA because it was too fragmented. To compile a reference set of assembly errors, each assembly is aligned to the reference genome using nucmer as described in Salzberg *et al*. (2012). Misassemblies were approximated by the alignment breaks, as most misassemblies will generate one or more breaks in the alignment (referred as NUCMER-Break). Gaps in the assemblies were excluded from consideration, as they do not necessarily represent assembly errors. Scaffolds that are smaller than 1 Kb were also excluded. Single-base differences were obtained by the show-snps command of the MUMmer package (referred to as NUCMER-SNV).

To obtain a list of regions identified by ALE as potentially containing errors, ALE per-base insert and placement scores were computed using a short fragment library against each of the six assemblies. Depth scores were not used, as Illumina sequencing data tends to be noisy. At each ALE score cutoff, the top-ranked bases were selected, and adjacent bases were merged into regions. If a region/base contained a NUCMER-SNV, or was within +/−50 bases centered at a NUCMER-Break, it was classified as a true positive for NUCMER-SNV or a NUCMER-Break, respectively. Otherwise, it was classified as 'Novel'. ALE can capture 15–70% of the break points at a stringent threshold (top 1000, Fig. 4A) among the six assemblies. At the same threshold, 40–78% of the ALE calls are classified as novel. These 'novel' sites are not necessarily false positives. As shown in Figure 4B, many of the 'novel' sites are actually SNPs, supported by the underlying short read data (colored vertical bars in the coverage track). Currently, ALE does not distinguish real SNPs from false positives.

3.7 ALE's performance with PacBio RS data

The aforementioned experiments were all performed with next generation short read data. Currently, the PacBio sequencing platform, also referred to as third-generation sequencing, is becoming increasingly popular owing to its long read length (up to several Kb) (Eid *et al*., 2009). These long reads are expected to greatly reduce the complexity associated with genome assembly validation. In contrast with second-generation sequencing, single-molecule-based PacBio RS sequencing has a much higher base error rate (~15%), making it an ideal candidate for testing the robustness of ALE against very noisy data. With this purpose, we examined the reference genome of Lambda Phage and corresponding PacBio reads of average depth 548× and a randomly sampled set at 50×. To determine ALE's performance on this dataset, the reference genome was synthetically mutated by adding 12 substitution, insertion and deletion errors at various locations (Supplementary Table S2). At 548× depth, within the top 12 worst placement sub-scores, ALE recovered all 12 errors at the mutated positions, while reporting no false positives. At 50× depth, excluding the low coverage edges, the 12 errors were detected in the top 14 worst placement sub-scores, with 2 false positives. In comparison, the standard Pacific Biosciences variant caller, EviCons, correctly identified only 10 of these errors with low confidence at default settings and the full 548× depth. This shows that ALE is a robust measure of assembly accuracy with noisy sequencing data and is a generic framework that can be used with both short and long sequence read technologies.

4 DISCUSSION

ALE facilitates the rapid discovery of many types of errors in genome assemblies including metagenomes. It does this by applying a rigorous statistical model, calculating the likelihood of observing a specific assembly, given the reads that were used to generate it, and calculating the contribution to this likelihood from each position in the assembly. This allows ALE to determine specific regions within a proposed assembly that are poorly supported by the reads. By integrating several aspects of the assembly and the reads, including k-mer composition, sequence depth, insert length and how well individual bases map, ALE is able to find errors as small as a single substitution error or indel, as well as large copy number errors and chimeric metagenome assemblies.

This framework can serve as a guide in optimizing genome assemblies in the following two ways. First, total ALE scores can be used to identify the best assembly from those generated by different assembly protocols. Second, by modifying the regions in which ALE reports low sub-scores, more accurate genomes can be constructed. The space of possible corrections to an input genome is too large to allow the current implementation of ALE to be used as an independent assembler, but it could be used to compare and combine the results from different assemblers and produce an assembly that is most likely to be correct. ALE could also be used to present an alternative method for

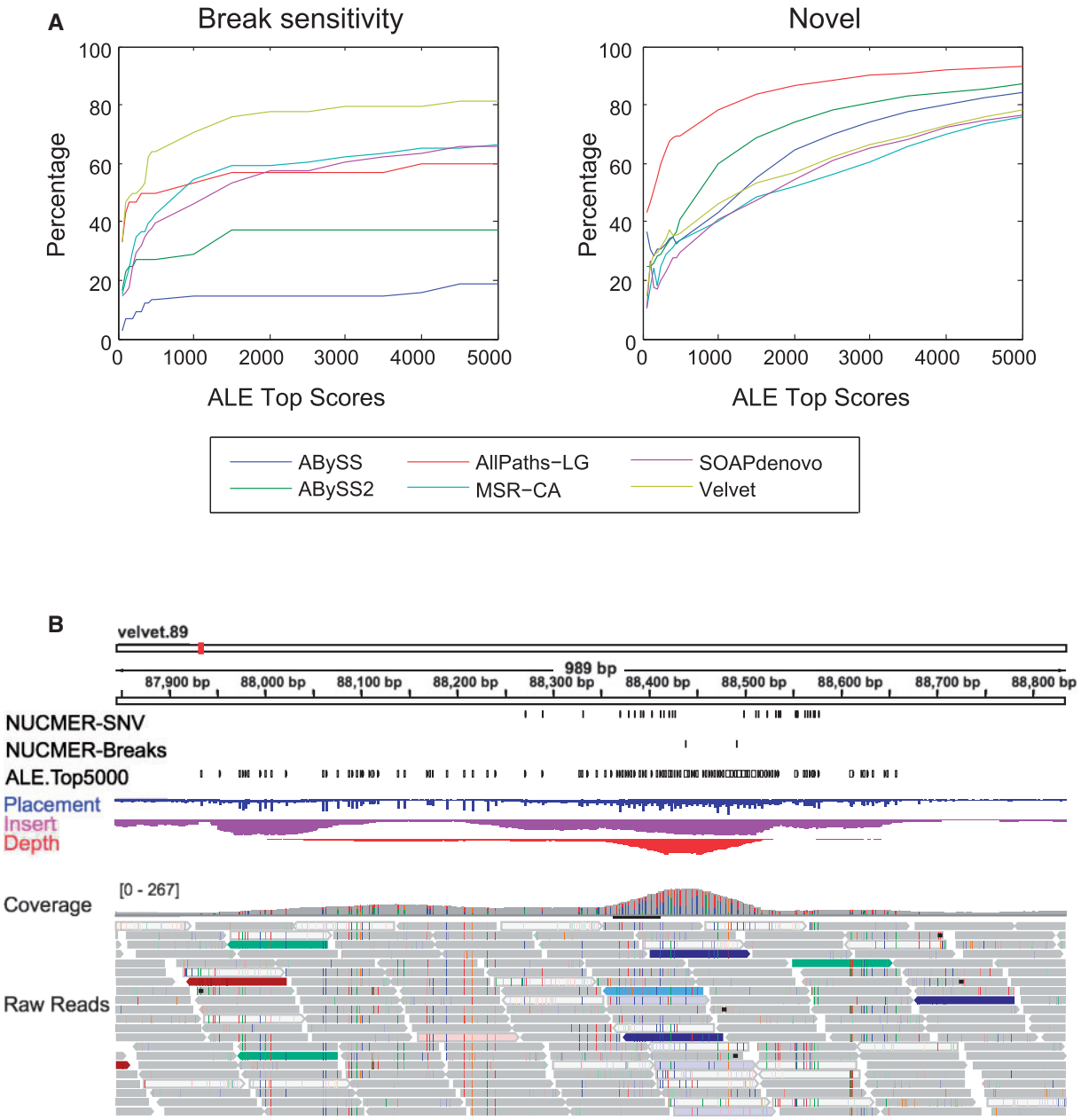

Fig. 4. ALE identifies regions with potential assembly errors. (**A**) Cumulative plots showing the percentage of assembly errors detected by ALE at different sensitivity thresholds. Detected assembly break points (break sensitivity), and novel calls (novel) at different ALE insert or placement thresholds (ALE Top Scores) for six assemblies of Staphylococcus using six different assemblers are shown. (**B**) A snapshot from Integrative Genomics Viewer for a scaffold from the velvet assembly of Staphlococcus. In the track Coverage, the height represents sequencing depth, and vertical colored bars represent potential SNPs. In the track Raw Reads, each gray horizontal bar represents a high quality aligned read, whereas horizontal color bars represent reads that may indicate problems (e.g. insert size is too big or too small). Vertical color bars are bases different from the reference sequence. More detailed description can be found at: http://www.broadinstitute.org/igv/AlignmentData

calculating assembly accuracy in local assembly algorithms such as Genovo (Laserson *et al.*, 2011).

When used with a reference genome and resequencing data, ALE can discover structural variations. As shown in the cases of *Spirochaeta smaragdinae* and *R. sphaeroides*, ALE readily detects structural variations whose sizes vary from a single base to several hundred kilobases.

ALE currently does not classify the type of assembly errors. Future work is needed to determine the profile of each type of assembly errors in a dataset-specific manner. Once ALE has this capability, it could guide an auto-correction algorithm to automatically fix problematic regions.

The effectiveness of ALE is influenced by the quality of its input: the read data and the alignments of those reads onto the

proposed assembly. Data with biased content or alignments, while accepted by ALE, tend to produce noisy sub-scores. The robustness of ALE, however, allows for the recovery of an accurate assembly accuracy measure as long as the random noise is consistent with the statistical model used by ALE (see Section 2).

The effectiveness of ALE is also influenced, like any Bayesian method, by the modeling assumptions implicit in its prior and likelihood. ALE's prior relies on the species-specific signature provided by tera-nucleotide frequencies (Teeling *et al*., 2004; Woyke *et al*., 2006), which enhances ALE's ability to detect contaminants from single genome assemblies and cross-assembly of genomes from related species, but may also lead to false positives in regions of horizontal gene transfer. Additionally, the insert score assumes that insert lengths are normally distributed, whereas other distributions may work better for some libraries. Future work could use an insert length distribution estimated non-parametrically from the data.

ALE could also be extended in future work to account for other factors that may currently lead to false positives, like origin of replication bias prevalent in circular genomes, horizontal gene transfer, automatic detection of sequencer bias and other distributions for insert length and coverage depth. Biases such as hard stops in Illumina could potentially be found by examining unlikely distributions of read orientation at specific locations coupled with low depth. Specific signatures within the different ALE metrics could be used to classify and correct for specific biases, much as ALE currently corrects for GC content (see Section 2).

ACKNOWLEDGEMENTS

The authors would like to thank Wendy Schackwitz for providing the *Rhodobacter* resequencing data and curated variants as well as Alicia Clum for assisting in the *Spirochaeta* data analysis. The authors would also like to thank the JGI Genome Technology Group for the use of their sequence data and expertise and Matthew Blow, Alex Copeland and Aleah Caulin for helpful edits and suggestions.

Funding: S.C. was supported by a Computational Science Graduate Fellowship, from the Office of Science of the U.S. Department of Energy under Contract No. DE-FG02-97ER25308. S.C. was also supported by a Startup and Production Allocation Award from the National Energy Research Scientific Computing Center (NERSC) of the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. The work conducted by the Department of Energy Joint Genome Institute was supported in part by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH112 and No. DE-AC02-05CH11231 (cow rumen metagenomics data analysis and informatics). P.F. was supported by Air Force Office of Scientific Research FA9550-12-1-0200.

Conflict of Interest: none declared.

REFERENCES

Aird,D. *et al*. (2011) Analyzing and minimizing bias in Illumina sequencing libraries. *Genome Biol*., **12**, R18.

Choi,J.H. *et al*. (2008) A machine learning approach to combined evidence validation of genome assemblies. *BMC Bioinformatics*, **24**, 744–750.

Choudhary,M. *et al*. (2006) Genome analyses of three strains of Rhodobacter sphaeroides: evidence of rapid evolution of chromosome II. *J. Bacteriol*., **189**, 1914–1921.

Darling,A. *et al*. (2011) Mauve assembly metrics. *Bioinformatics*, **27**, 2756–2757.

Durfee,T. *et al*. (2008) The complete genome sequence of Escherichia coli DH10B: insights into the biology of a laboratory workhorse. *J. Bacteriol*., **190**, 2597–2606.

Earl,D. *et al*. (2011) Assemblathon 1: a competitive assessment of de novo short read assembly methods. *Genome Res*., **21**, 2224–2241.

Eid,J. *et al*. (2009) Real-time DNA sequencing from single polymerase molecules. *Science*, **323**, 133–138.

Fujimoto,A. *et al*. (2010) Whole-genome sequencing and comprehensive variant analysis of a Japanese individual using massively parallel sequencing. *Nat. Genet*., **42**, 931–936.

Haiminen,N. *et al*. (2011) Evaluation of methods for de novo genome assembly from high-throughput sequencing reads reveals dependencies that affect the quality of the results. *PLoS ONE*, **6**, e24182.

Hess,M. *et al*. (2011) Metagenomic discovery of biomass-degrading genes and genomes from cow rumen. *Science*, **331**, 463–467.

Iverson,V. *et al*. (2012) Untangling genomes from metagenomes: revealing an uncultured class of marine euryarchaeota. *Science*, **335**, 587–590.

Kent,J.W. *et al*. (2002) The human genome browser at UCSC. *Genome Res*., **12**, 996–1006.

Lander,E.S. and Waterman,M.S. (1988) Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics*, **2**, 231–239.

Langmead,B. *et al*. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol*., **10**, R25.

Laserson,J. *et al*. (2011) Genovo: de novo assembly for metagenomes. *J. Comput. Biol*., **18**, 429–443.

Li,H. *et al*. (2009) The sequence alignment/map (SAM) format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Li,R. *et al*. (2010) De novo assembly of human genomes with massively parallel short read sequencing. *Genome Res*., **20**, 265–272.

Lin,Y. *et al*. (2011) Comparative studies of de novo assembly tools for next-generation sequencing technologies. *Bioinformatics*, **27**, 2031–2037.

Mavromatis,K. *et al*. (2010) Complete genome sequence of Spirochaeta smaragdinae type strain. *Stand. Genomic Sci*., **3**, 136–144.

Meader,S. (2010) Genome assembly quality: assessment and improvement using the neutral indel model. *Genome Res*., **20**, 675–684.

Metzker,M.L. (2010) Sequencing technologies - the next generation. *Nat. Rev. Genet*., **11**, 31–46.

Narzisi,G. and Mishra,B. (2011) Comparing de novo genome assembly: the long and short of it. *PLoS One*, **6**, e19175.

Nicol,J.W. *et al*. (2009) The integrated genome browser: free software for distribution and exploration of genome-scale datasets. *Bioinformatics*, **25**, 2730–2731.

Phillippy,A. *et al*. (2008) Genome assembly forensics: finding the elusive mis-assembly. *Genome Biol*., **9**, R55.

Pop,M. (2009) Genome assembly reborn: recent computational challenges. *Brief. Bioinform*., **10**, 354–366.

Qin,J. *et al*. (2010) A human gut microbial gene catalogue established by metagenomic sequencing. *Nature*, **464**, 59–65.

Salzberg,S.L. *et al*. (2012) GAGE: a critical evaluation of genome assemblies and assembly algorithms. *Genome Res*., **22**, 557–567.

Schmutz,J. *et al*. (2010) Genome sequence of the palaeopolyploid soybean. *Nature*, **463**, 178–183.

Teeling,H. *et al*. (2004) Application of tetranucleotide frequencies for the assignment of genomic fragments. *Environ. Microbiol*., **6**, 938–947.

Vezzi,F. *et al*. (2012) Feature-by-feature–evaluating de novo sequence assembly. *PloS One*, **7**, e31002.

Wang,W. *et al*. (2011) Next generation sequencing has lower sequence coverage and poorer SNP-detection capability in the regulatory regions. *Sci. Rep*., **1**, 55.

Woyke,T. *et al*. (2006) Symbiosis insights through metagenomic analysis of a microbial consortium. *Nature*, **443**, 950–955.

Woyke,T. *et al*. (2010) One bacterial cell. One complete genome. *PLoS One*, **5**, e10314.

Wu,D. *et al*. (2009) A phylogeny-driven genomic encyclopaedia of Bacteria and Archaea. *Nature*, **462**, 1056–1060.

Yilmaz,P. *et al*. (2011) Minimum information about a marker gene sequence (MIMARKS) and minimum information about any (x) sequence (MIxS) specifications. *Nat. Biotechnol*., **29**, 415–420.

Zimin,A.V. *et al*. (2008) Assembly reconciliation. *BMC Bioinformatics*, **24**, 42–45 .