metaMIC: reference-free Misassembly Identification and Correction of *de novo* metagenomic assemblies

Senying Lai[1], Shaojun Pan[1], Luis Pedro Coelho[1,4,§], Wei-Hua Chen[2,3,§], Xing-Ming Zhao[1,4,5,§]

Affiliations

[1]Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, Shanghai 200433, China.

[2]Key Laboratory of Molecular Biophysics of the Ministry of Education, Hubei Key Laboratory of Bioinformatics and Molecular-imaging, Center for Artificial Intelligence Biology, Department of Bioinformatics and Systems Biology, College of Life Science and Technology, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China.

[3]College of Life Science, Henan Normal University, Xinxiang, Henan 453007, China.

[4]MOE Key Laboratory of Computational Neuroscience and Brain-Inspired Intelligence, and MOE Frontiers Center for Brain Science, Shanghai 200433, China.

[5]Research Institute of Intelligent Complex Systems, Fudan University, Shanghai 200433, China.

[§]To whom correspondence should be addressed.

Emails

Senying Lai: 19110850024@fudan.edu.cn

Shaojun Pan: 19110850020@fudan.edu.cn

Luis Pedro Coelho: luis@luispedro.org

Wei-Hua Chen: weihuachen@hust.edu.cn

Xing-Ming Zhao: xmzhao@fudan.edu.cn

Abstract

Evaluating the quality of metagenomic assemblies is important for constructing reliable metagenome-assembled genomes and downstream analyses. Here, we present metaMIC (https://github.com/ZhaoXM-Lab/metaMIC), a machine-learning based tool for identifying and correcting misassemblies in metagenomic assemblies. Benchmarking results on both simulated and real datasets demonstrate that metaMIC outperforms existing tools when identifying misassembled contigs. Furthermore, metaMIC is able to localize the misassembly breakpoints, and the correction of misassemblies by splitting at misassembly breakpoints can improve downstream scaffolding and binning results.

Keywords

Metagenomic assemblies, Misassembled contigs, Misassembly breakpoints, Metagenome-assembled genomes, Binning

Background

Constructing reliable metagenome-assembled genomes (MAGs) is of great importance for understanding microbial communities and downstream functional analysis, such as taxonomic annotations and reconstruction of metabolic processes [1-4]. MAGs are obtained by binning assembled contigs into bins, the quality of which can be significantly affected by the assembly errors in contigs. For example, the chimerical assemblies consisting of two or more genomes can introduce contamination for reconstructed MAGs, potentially resulting in misleading biological conclusions [5]. Despite the progress in assembly algorithms, errors are still prevalent in metagenomic-assembled contigs owing to the inherent complexity of metagenomic data. Assembly errors including inter- and intra-species misassemblies are caused by

repetitive genomic regions that occur within the same genome or conserved sequences shared among distinct organisms, which is especially likely to happen when multiple closely-related strains are present in the same environment [6, 7]. Therefore, the evaluation of metagenomic assemblies is critical for constructing high-quality and reliable MAGs.

Approaches that have been proposed for assessing the quality of metagenomic assemblies can be grouped into two categories: *reference-based* and *reference-free* approaches. Reference-based methods evaluate the *de novo* assemblies by aligning them against corresponding reference genomes. For example, MetaQUAST [8], the metagenomic-adapted version of QUAST [9], detects misassemblies such as translocation, inversion and relocation by mapping the metagenomic contigs to a set of closely-related reference genomes. However, it is difficult to distinguish errors from real structural variation. Moreover, reference genomes are available for only a small fraction of organisms found in real environments, which limits these approaches to previously-sequenced species [10]. Therefore, the evaluation of metagenomic assemblies would benefit from reference-free methods. Typically these methods exploit features such as the high variation in coverage depth or inconsistent insert distance of paired-end reads to indicate possible repeat collapse, misjoins or error insertions/deletions [11]. Popular reference-free methods include ALE [12], DeepMAsED [13], SuRankCo [14] and VALET [15]. ALE measures the quality of assemblies as the likelihood that the observed reads are generated from a given assembly by modeling the sequencing process. SuRankCo uses a machine learning approach to provide quality scores for contigs based on characteristics of contigs such as length and coverage. VALET detects misassemblies based on the combination of multiple metrics extracted from the alignment of reads to contigs. DeepMAsED employs a deep learning approach to identify misassembled contigs. Despite the great value of those approaches

for evaluating metagenomic assembly quality, only VALET and ALE predict the position where the misassembly errors are introduced and none of these methods have functionality for correcting metagenomic misassemblies. More importantly, VALET and SuRankCo are no longer maintained, and software incompatibilities hinder their use.

Here, we present a novel tool called metaMIC which performs reference-free misassembly identification and correction in *de novo* metagenomic assemblies. metaMIC can identify misassembled contigs, localize misassembly breakpoints within misassembled contigs and then correct misassemblies by splitting misassembled contigs at breakpoints. Benchmarking results on both simulated and real metagenomics data show that metaMIC can identify misassembled contigs with higher accuracy than state-of-the-art tools, and precisely localize the misassembly error regions and recognize breakpoints in both single genomic and metagenomic assemblies. By comparing the scaffolding and binning results before and after metaMIC correction, we demonstrate that the correction of misassemblies by metaMIC can improve the scaffolding and binning results.

Results

Overview of the metaMIC pipeline

metaMIC is a fully automated tool for identifying and correcting misassemblies in metagenomic contigs using the following three steps (Fig. 1). First, various types of features were extracted from the alignment between paired-end sequencing reads and each contig, including sequencing coverage, nucleotide variants, mate pair consistency, and *k*-mer abundance differences (KAD) [16] between mapped reads and the contig. The KAD method was previously developed for evaluating the accuracy of nucleotide base

quality in single genomic assemblies. Here, we extended KAD to metagenomic assemblies to measure the overall consistency between mapped reads and corresponding contigs (see Methods). Secondly, the features extracted in the first step are used as input to a random forest classifier for identifying misassembled contigs, where the classifier is trained with simulated bacterial metagenomic communities to discriminate misassembled contigs from correctly assembled ones. Thirdly, metaMIC will localize misassembly breakpoint(s) in each misassembled contig, namely the point at which the left and right flanking sequences are predicted to have originated from different genomes or locations. As most misassemblies are chimeras where two fragments from different locations or with different orientations are mistakenly connected and not just random sequences being generated [9], misassemblies can be corrected by breaking up the contigs into two (or more) correctly assembled contigs.

Identifying misassembled contigs in simulated metagenomic datasets

To evaluate metaMIC, we tested it on simulated metagenomic datasets obtained from CAMI (the Critical Assessment of Metagenome Interpretation) [2] that comprise a known mixture of organisms. We first evaluated metaMIC on the Medium (*CAMI1-Medium*) and High-diversity communities (*CAMI1-High*) to see how dataset complexity will influence the accuracy of metaMIC. We noticed that the types of misassemblies identified in these two datasets were slightly different, and the CAMI1-High dataset contains more inter-species translocations and higher proportion of misassemblies while the CAMI1-Medium dataset contains more relocations (see Figs. S1, S2), which is consistent with previous conclusion that datasets with higher intra-species diversity tend to have more inter-species translocation misassemblies [13]. Compared with CAMI1-High metaMIC performed better on (Fig. 2a; although still

112 significantly better than existing tools) CAMI1-Medium, implying that the higher microbial diversity
113 increases the challenge of identifying misassembled contigs. We further compared metaMIC on these
114 datasets against ALE [12] and DeepMAsED [13] (See Methods). As shown in Fig. 2a, metaMIC
115 significantly outperforms both ALE and DeepMAsED on the two datasets, as MetaMIC achieved 4-fold
116 higher AUPRC (area under the precision-recall curve).

117 We also evaluated metaMIC and other tools on simulated metagenomic datasets from three different
118 human body sites: gastrointestinal tract (*CAMI2-Gut*), skin (*CAMI2-Skin*) and oral cavity (*CAMI2-Oral*).
119 As shown in Figs. 2b, c and Fig. S3, metaMIC has the highest precision when identifying misassembled
120 contigs at any recall threshold. Additionally, we tested metaMIC on a simulated virome datasets (*Sim-*
121 *Virome*), which were simulated based on 1,000 complete viral genomes randomly selected from NCBI
122 RefSeq collection [17] (See Methods). The Sim-Virome contains mainly translocations and relocations
123 but few inter-species translocations and inversions. We found that metaMIC still significantly
124 outperforms both ALE and DeepMAsED on Sim-Virome dataset as shown in Fig. 2d, indicating that
125 metaMIC can also be used for virome assemblies besides bacterial metagenomic assemblies.

126 As metaMIC can be trained on contigs assembled by different assemblers, we further investigated
127 the impact of different assemblers on the performance of metaMIC when identifying misassembled
128 contigs. Here, two popular assemblers, i.e. MEGAHIT and IDBA_UD, used for metagenomic data were
129 considered. As shown in Fig. 2e, we found that metaMIC performed best when it was trained on the same
130 assembler as it was later evaluated. Therefore, we recommend to use metaMIC trained on the contigs
131 generated by the same assembler. For version 1.0, metaMIC provides builtin models supporting

MEGAHIT and IDBA_UD as well as the ability to generate new models based on the assembler specified by users.

metaMIC can identify breakpoints with higher accuracy in misassembled contigs

Beyond identifying misassembled contigs, metaMIC is able to accurately recognize the misassembly breakpoints, at which the misassembled contigs can be split into shorter ones. From the distribution, we can clearly see that the error regions containing any misassembly type generally have significantly higher anomaly scores than error-free regions, and the inter-species translocation error is most prevalent in the dataset. In the CAMI datasets, it is indeed the inter-species translocation error that occurs most often (Fig. S2). The differential distribution of anomaly scores between error and error-free regions implies that the anomaly score has the potential to recognize the error regions. We also noticed that the misassembly sites are usually read breakpoints (locations at which the boundaries of aligned read fragments do not coincide with the ends of corresponding reads) [18]. Similar to anomaly scores, we found that the read breakpoint ratio was significantly different between error regions and error-free regions (Fig. 3b, see also Figs. S7, S8).

Due to the potential of read breakpoint ratio and anomaly score to localize the error regions, we want to see whether metaMIC can use these two features to separate the erroneous regions from error-free regions. From the receiver operation curves shown in Fig. 3c, we can see that with either anomaly score or read breakpoint ratio, metaMIC can classify the error regions containing misassembly breakpoints with error-free regions more accurately than ALE. To combine the usages of these two

features, metaMIC first localizes the error regions in a misassembled contigs with the help of anomaly score, and then identifies the exact breakpoints in an error region based on the read breakpoint ratio.

We evaluated the performance of both metaMIC and ALE on the five datasets from CAMI as shown in Fig. 3d. We observed that approximately 71-86% of the metaMIC-predicted breakpoints were within 500bp compared to 26-48% of those by ALE. More importantly, metaMIC could predict the exact locations for ~25% of the breakpoints with the use of read breakpoints. Again, inter-species translocations or inversions can be detected with higher accuracy relative to other misassembly types (Fig. 3e), consistent with previous results that they were supported by more fragmentally aligned reads and had higher anomaly scores as compared with other error types (see Figs. 3a, b; Fig. S8). Given the possible influence of contig length on the prediction error size, we normalized the error size by the contig length, and compared the results of metaMIC with those of ALE. As shown in Fig. 3f, metaMIC still significantly outperforms ALE with respect to the normalized error size (Wilcoxon test, p-value <2.22e-16), where the median and mean of the metaMIC's normalized error size were 0.01 and 0.11, respectively, compared to 0.39 and 0.34 for ALE (see also Fig. S9).

Splitting misassembled contigs improves downstream binning performance

As metaMIC can identify breakpoints in misassembled contigs, it can split misassembled contigs at breakpoints and reduce the number of misassemblies (see Methods); although the contiguity could be slightly decreased due to more fragmented contigs [19]. To see how the correction of splitting misassembled contigs at breakpoints employed by metaMIC will influence downstream analyses, we binned the contigs in the simulated datasets. We then assessed the binning performance over the original

and metaMIC-corrected contigs by counting the number of obtained high-quality bins. We can see in Fig. 4a that metaMIC correction increases the number of near-complete reconstructed bins (completeness above 90%, contamination below 5% [3]) by 10-20% (see also Fig. S13a, Table S1), showing that the correction of metagenomic miassemblies has significant impact on downstream binning. We noticed that most of the misassemblies corrected by metaMIC were inter-species translocations that were also the main sources of chimeras and assembly errors in CAMI datasets (Fig. S2, Table S2). From Fig. 4b and also those shown in Fig. S13b, we can see that bin-wise F1 scores of those bins constructed from corrected contigs are significantly improved compared with the results over original contigs, indicating that the reconstructed bins can better represent the reference genomes after metaMIC correction. The above results clearly demonstrate that the correction of metagenomic misassemblies by metaMIC can significantly improve the resulting bins in term of both completeness and contaminations, which is important for understanding the complex microbiota communities.

Application of metaMIC to real metagenomic datasets

To better evaluate the performance of metaMIC, we applied metaMIC to two recent human gut metagenomics datasets from Ethiopian [20] and Madagascar [21] cohorts that consist of 50 and 112 samples, respectively. In total, metaMIC respectively identified 5,905 and 18,436 misassembled contigs in *Ethiopian* and *Madagascar* datasets, which represents 2.59% and 4.53% of all contigs in the two datasets. We then separately binned the original and corrected contigs into bins. Strikingly, we found that ~20% of the resulting original-bins contained misassemblies, although the latter accounted for less than 5% of all contigs (See Table S3). As previous results have demonstrated that metaMIC correction

194 can improve the binning results in simulated datasets, we further explored whether the correction step
195 employed by metaMIC can improve downstream results in real datasets. As shown in Fig. 5, in addition
196 to obtaining more bins of high quality (Completeness >90 and Contamination <5) (Fig. 5a, Table S3),
197 the corrected bins had an equal or higher F1 score compared to the corresponding original bins (Fig. 5b).
198 The results indicate that the misassembled contigs identified by metaMIC in these two real datasets are
199 really misassembled, the correction of which can significantly improve downstream analysis results.

200 As these contamination metrics are based on *in silico* evaluation, we further tested the ability of
201 identifying misassemblies using another metagenomic dataset (a combined rumen fluid and solid sample)
202 where both short and long reads are available. Since the long reads from PacBio platforms are able to
203 span repeats [22, 23], which are the main contributor to misassemblies, we can therefore use the long-
204 read assemblies as gold standards to validate our predicted misassembled contigs from the short-read
205 assemblies. In total, metaMIC identified 692 misassembled contigs (approximately 2.5%) in the short-
206 read assemblies. By manual inspection of the alignments between PacBio assemblies and short-read
207 assemblies, we can validate a subset of metaMIC predictions (Fig. 5c and Fig. S14). For instance, there
208 exist two peaks at positions of 1200bp and 6920bp in the contig of "k141_847840" according to the
209 anomaly scores by metaMIC, and both peaks, especially the one at 6920bp, contain higher read
210 breakpoint counts implying possible misassembly breakpoints at these two locations. When aligning this
211 contig against the long-read assemblies, we found that two regions in this contig (1201-6738bp and 6920-
212 8700bp) were indeed aligned to two different long-read assembled contigs, and a change-point in the
213 read coverage at 6920bp can be observed (see Fig. 5c), indicating that there are actually two contigs

214 wrongly assembled into one contig at position of 6920bp. We also found that only a few reads can be
215 aligned to the region of 0-1200bp, suggesting this region may be extended mistakenly by the assembler.
216

217 **Application of metaMIC to isolate genomes**

218 Since metaMIC can identify and correct intra-species misassemblies such as inversions and relocations,
219 metaMIC can also be applied to isolate genomes. We tested metaMIC on four real datasets from GAGE-
220 B project [24], which aimed to evaluate assembly algorithms on isolate genomes. We tested metaMIC
221 on *B. cereus*, *M. abscessus*, *R. sphaeroides* and *V. cholerae*, where the raw reads, assembled contigs [25]
222 and curated reference genomes are available for these four species. metaMIC was ran on the assemblies
223 downloaded from GAGE-B project and its performance was evaluated with the results by QUAST [9] as
224 gold standard. These four datasets contain mainly relocations but a few translocations (Table S4). We
225 noticed that similar to metagenomes, the error regions in isolate genomes also have higher anomaly
226 scores and more read breakpoints than error-free regions (Fig. S15). We then compared metaMIC against
227 MEC [26], a recently-developed misassembly correction tool, when identifying misassembly breakpoints
228 on the four isolate genomes. As shown in Table 2, metaMIC identified more true misassemblies than
229 MEC, where approximately 80% misassemblies can be corrected compared to ~30% of MEC; and after
230 the correction by metaMIC, the total number of bases of uncorrected misassembled contigs (i.e.
231 misassembled contig length in Table 2) was significantly reduced compared with that by MEC.

232 To further see influence of misassembly correction on isolate genomes, we scaffolded original and
233 corrected contigs separately with popular scaffolders including BESST [27] and ScaffMatch [28], and
234 then used QUAST to evaluate the scaffolding results. As seen in Table 3 and supplementary Table 6, the

number of misassemblies in the scaffolding results based on metaMIC's corrected contigs was much lower than that based on the original uncorrected contigs, and metaMIC significantly outperforms MEC in terms of misassembled contig length. Moreover, metaMIC performs comparably well or better compared against MEC in terms of NA50 and total aligned length, and performs better especially for *R.sphaeroides*. The above results clearly show the effectiveness of metaMIC when identifying and correcting misassembled contigs on isolate genomes, and also the capability of maintaining or improving the contiguity of downstream scaffolding after correction.

Discussion

We present a novel tool named metaMIC to identify and correct misassembled contigs from *de novo* metagenomic assemblies and demonstrate its effectiveness on both simulated and real metagenomic datasets of varying complexity. Unlike most existing metagenomic assembly evaluation methods that only evaluate individual contigs or overall assemblies, metaMIC is capable of localizing the misassembly breakpoints and then corrects the misassembled contigs at breakpoints. By integrating various types of features extracted from both reads and assemblies, including read coverage, mate pair consistency, nucleotide variants and *k*-mer abundance consistency, metaMIC is able to detect intra- and inter-species misassemblies. Additionally, metaMIC can also be applied on isolate genomes given its ability in identifying intra-species misassemblies. After the correction of misassemblies, metaMIC can significantly help improve the performance of downstream analysis including binning and scaffolding.

In this study, the performance of metaMIC is mainly shown on the metagenomic assemblies assembled by MEGAHIT due to its high memory efficiency [29]. As different assemblers tend to be

biased to certain types of misassemblies, the models trained on the outputs of one assembler may not transfer well to other assemblers. Note that metaMIC can be easily extended to work on the metagenomic assemblies by other assembler tools if the training datasets generated by the corresponding assemblers are available. We suggest to use metaMIC on the datasets from the same assembler as the one it is trained on.

metaMIC scans each contig with a sliding window of 100bp to localize the candidate error regions. Generally, a shorter window size can have a higher resolution to pinpoint error regions but require more computation resources while the longer window size can be robust to noise but are more likely to cover multiple errors. In addition, metaMIC currently cannot distinguish the types of assembly errors. In the future, more work is needed to determine the error types which in turn can help to correct misassemblies more accurately.

metaMIC correction mainly relies on splitting contigs at misassembly breakpoints. However, caution should be needed here as more fragmented sequences will be generated and mistakenly splitting may result in disrupted gene structure, which can have adverse influence on downstream functional genomic analysis. Although we have showed that metaMIC correction can improve the downstream binning results, the quality of reconstructed draft bins can be further improved if the broken contigs can be joined into scaffolds correctly. Thus, the combination of metaMIC and scaffolding algorithms will be a promising direction for future research, leading to effective approaches for reconstructing genomes from sequencing data with higher quality and completeness.

Several directions hold promise for further improvements to metaMIC. Firstly, metagenomic read mapping can be evaluated in more robust manner by aligning multi-assigned reads in a probabilistic

277 manner to their contig of origin [30] or using base-level quality metrics such as CIGAR strings [31].
278 Secondly, increasing the amount of training data and integrating other assemblers such as metaSPAdes
279 [32] are also potential directions for the improvement of metaMIC. Thirdly, the factors that may result
280 in false positive predictions, such as structural variation within species of high similarity and G-C bias
281 in sequencing coverage could be taken into account in future work. Finally, as reference genomes of
282 many bacterium are available, a better performance can be achieved by the combination of reference-
283 free and reference-based approaches.
284

285 # Conclusions

286 Here, a novel tool named metaMIC is developed for identifying and correcting misassemblies in *de novo*
287 metagenomic assemblies without the use of reference genomes. Benchmarking on both simulated and
288 real datasets, we show that metaMIC is able to pinpoint misassemblies in both single and metagenomic
289 assemblies. We also demonstrate that metaMIC is able to improve the scaffolding or binning results by
290 splitting misassembled contigs at misassembly breakpoints. As none of current assemblers can achieve
291 a completely accurate assembly and misassemblies in contigs have negative influence on downstream
292 analysis, we expect metaMIC can serve as a guide in optimizing metagenomic assemblies and help
293 researchers be aware of problematic regions in assembled contigs, so as to avoid misleading downstream
294 biological analysis.
295

296 # Methods

297 ## metaMIC workflow

metaMIC is implemented in Python3 (Python $\geq$ 3.6). It requires assembled contigs in FASTA format and paired-end reads in FASTA or FASTQ format as input. Alternatively, the user can provide a BAM file with read pairs mapping to contigs. Given the contigs, metaMIC will first identify the misassemblies by employing a random forest classifier trained on the features extracted from reads and contigs. Next, metaMIC will identify the regions containing misassembly breakpoints in the misassembled contigs based on the anomaly scores, and then recognize the exact positions of the breakpoints in the error regions. Then metaMIC will correct the misassemblies by splitting the contigs at the breakpoints. The details will be given below.

Features extracted from reads and contigs

BWA-MEM (v.0.7.17) [33] is used to map paired-end reads to assemblies, followed by using samtools (v1.9) [34] to filter low quality mappings and sorting the alignments. Then four types of features will be extracted from the sorted BAM file, including read coverage, mate-pair consistency, nucleotide variants and *k*-mer abundance difference.

For each paired-end reads with left and right mate reads, the insert size corresponding to the distance between two mates is assumed to follow normal distribution [26]. A read is regarded as a proper read if the insert size belongs to $[\mu - 3\sigma, \mu + 3\sigma]$ and the orientation is consistent with its mate, and is a discordant read otherwise. A read is regarded as a clipped read if it contains at least 20 unaligned bases at either end of the read, and a read is regarded as a supplementary read if different parts of the read are aligned to different regions of contigs.

The coverage-based features include standardized read coverage, fragment coverage and their deviation. The read coverage per base represents the number of reads that are mapped over that base, and the fragment coverage is the number of proper paired-end reads spanning that base. The read coverage and fragment coverage are further standardized as the ones divided by the means of the corresponding coverages of all bases across the contig or a given region.

The nucleotide variants information is extracted from BAM file with the help of samtools. metaMIC counts the number of discordant, ambiguous and correct alignments separately at each position. For each type of alignment in a contig, metaMIC will calculate the proportion of the alignment by dividing the number of this type of alignments to the total number of mapped bases across the contig, and the same for a given region.

metaMIC calculates the *k*-mer abundance difference (KAD) at each base based on the alignment of paired-end reads to contigs. The KAD value, proposed by He et al [16], measures the consistency between the abundance of a *k*-mer from short reads and the occurrence of the *k*-mer in the genome. A *k*-mer with KAD value not belonging to [-0.5, +0.5] will be regarded as an error *k*-mer, and a base is regarded as an error base if an error *k*-mer covers that base. For a given contig, metaMIC will count the number of error bases across the contig and divide it by the contig length. The proportion of error bases within a given region from a contig will be calculated in the same way.

In summary, the above these four types of features will be extracted for the whole contig (contig-based features) or a window of 100bp (window-based features). The contig-based features will be used to train a random forest to identify misassembled contigs, while the window-based features will be used as input of isolate forests to recognize the error regions containing breakpoints.

Identification of misassembled contigs

With the above contig-based features, metaMIC trains a random forest [35] implemented in Scikit-Learn [36] to discriminate misassembled contigs from those correctly assembled ones, where an ensemble of 1,000 trees are used. For each contig, a probability score representing the likelihood that the contig is misassembled will be output by metaMIC. The random forest model was trained on a training dataset containing contigs assembled from simulated bacterial metagenomes, whereas the ground truth misassembly labels of contigs provided MetaQUAST are used as a target for training the model. Due to the existence of strong class imbalance, we down-sampled the training dataset to obtain the same number of correct contigs paired with the misassembled contigs.

Localizing breakpoints in misassembled contigs

After identifying misassembled contigs, metaMIC is able to localize the misassembly breakpoints in those misassembled contigs. Firstly, metaMIC scans each contig with a sliding window of 100bp, and calculates an anomaly score for each window by employing isolation forest [37] based on window-based features to localize the error regions containing misassembly breakpoints, where the region with a higher anomaly score may be an error region; Secondly, metaMIC uses the read breakpoint ratio to recognize the exact misassembly breakpoint in an error region. Specifically, for a given predicted misassembled contig, metaMIC identifies a 100bp region with the highest anomaly score as an error region and then the position with the highest read breakpoint ratio within this window as the misassemly breakpoint. For those error regions without read breakpoints, the central position of the error region is regarded as the misassembly breakpoint.

Evaluation of binning results

When evaluating a set of bins reconstructed from simulated microbial datasets, we use BLASTn to map each bin against the ground truth genomes used for each dataset. A representative genome of each bin is determined based on the genome which can be covered by the highest fraction of nucleotides from that bin. Then for each bin, we define the number of nucleotides in the bin that belong to the representative genome as true positives (TP). The total number of nucleotides from the bin not covered by the representative genome corresponds to the false positives (FP), and the number of nucleotides in the representative genome not covered by any contigs from that bin represents the false negatives (FN). Then the completeness, contamination and F1 score of each bin can be calculated as follows.

$$completeness = \frac{TP}{TP + FN}$$

$$purity = \frac{TP}{TP + FP}$$

$$contamination = 1 - purity$$

$$F1\ score = \frac{2 * completeness * purity}{completeness + purity}$$

For the real metagenomics data sets where the ground truth genomes are inaccessible, we employ CheckM [38] to estimate the completeness and contamination of each bin.

Abbreviations

MAG: metagenome-assembled genomes

bp: base pair

AUPRC: area under the precision-recall curve

KAD: *k*-mer abundance difference

References

1. Salzberg SL, Phillippy AM, Zimin A, Puiu D, Magoc T, Koren S, Treangen TJ, Schatz MC, Delcher AL, Roberts M, et al: **GAGE: A critical evaluation of genome assemblies and assembly algorithms.** *Genome Res* 2012, **22:**557-567.
2. Sczyrba A, Hofmann P, Belmann P, Koslicki D, Janssen S, Droge J, Gregor I, Majda S, Fiedler J, Dahms E, et al: **Critical Assessment of Metagenome Interpretation-a benchmark of metagenomics software.** *Nat Methods* 2017, **14:**1063-1071.
3. Bowers RM, Kyrpides NC, Stepanauskas R, Harmon-Smith M, Doud D, Reddy TBK, Schulz F, Jarett J, Rivers AR, Eloe-Fadrosh EA, et al: **Minimum information about a single amplified genome (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and archaea.** *Nat Biotechnol* 2017, **35:**725-731.
4. Shaiber A, Eren AM: **Composite Metagenome-Assembled Genomes Reduce the Quality of Public Genome Repositories.** *mBio* 2019, **10**.
5. Chen LX, Anantharaman K, Shaiber A, Eren AM, Banfield JF: **Accurate and complete genomes from metagenomes.** *Genome Res* 2020, **30:**315-333.
6. Kingsford C, Schatz MC, Pop M: **Assembly complexity of prokaryotic genomes using short reads.** *BMC Bioinformatics* 2010, **11:**21.
7. Nagarajan N, Pop M: **Parametric complexity of sequence assembly: theory and applications to next generation sequencing.** *J Comput Biol* 2009, **16:**897-908.
8. Mikheenko A, Saveliev V, Gurevich A: **MetaQUAST: evaluation of metagenome assemblies.** *Bioinformatics* 2016, **32:**1088-1090.
9. Gurevich A, Saveliev V, Vyahhi N, Tesler G: **QUAST: quality assessment tool for genome assemblies.** *Bioinformatics* 2013, **29:**1072-1075.
10. Rusch DB, Halpern AL, Sutton G, Heidelberg KB, Williamson S, Yooseph S, Wu D, Eisen JA, Hoffman JM, Remington K, et al: **The Sorcerer II Global Ocean Sampling expedition: northwest Atlantic through eastern tropical Pacific.** *PLoS Biol* 2007, **5:**e77.
11. Zhu X, Leung HC, Wang R, Chin FY, Yiu SM, Quan G, Li Y, Zhang R, Jiang Q, Liu B, et al: **misFinder: identify mis-assemblies in an unbiased manner using reference and paired-end reads.** *BMC Bioinformatics* 2015, **16:**386.
12. Clark SC, Egan R, Frazier PI, Wang Z: **ALE: a generic assembly likelihood evaluation framework for assessing the accuracy of genome and metagenome assemblies.** *Bioinformatics* 2013, **29:**435-443.
13. Mineeva O, Rojas-Carulla M, Ley RE, Scholkopf B, Youngblut ND: **DeepMAsED: evaluating the quality of metagenomic assemblies.** *Bioinformatics* 2020, **36:**3011-3017.
14. Kuhring M, Dabrowski PW, Piro VC, Nitsche A, Renard BY: **SuRankCo: supervised ranking of contigs in de novo assemblies.** *BMC Bioinformatics* 2015, **16:**240.
15. Olson ND, Treangen TJ, Hill CM, Cepeda-Espinoza V, Ghurye J, Koren S, Pop M: **Metagenomic assembly through the lens of validation: recent advances in assessing and improving the quality of genomes assembled from metagenomes.** *Brief Bioinform* 2019, **20:**1140-1150.
16. He C, Lin G, Wei H, Tang H, White FF, Valent B, Liu S: **Factorial estimating assembly base**

errors using k-mer abundance difference (KAD) between short reads and genome assembled sequences. *NAR Genom Bioinform* 2020, **2:**lqaa075.

17. Pruitt KD, Tatusova T, Maglott DR: **NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins.** *Nucleic Acids Res* 2007, **35:**D61-65.

18. Phillippy AM, Schatz MC, Pop M: **Genome assembly forensics: finding the elusive mis-assembly.** *Genome Biol* 2008, **9:**R55.

19. Hunt M, Kikuchi T, Sanders M, Newbold C, Berriman M, Otto TD: **REAPR: a universal tool for genome assembly evaluation.** *Genome Biol* 2013, **14:**R47.

20. Pasolli E, Asnicar F, Manara S, Zolfo M, Karcher N, Armanini F, Beghini F, Manghi P, Tett A, Ghensi P, et al: **Extensive Unexplored Human Microbiome Diversity Revealed by Over 150,000 Genomes from Metagenomes Spanning Age, Geography, and Lifestyle.** *Cell* 2019, **176:**649-662 e620.

21. Almeida A, Mitchell AL, Boland M, Forster SC, Gloor GB, Tarkowska A, Lawley TD, Finn RD: **A new genomic blueprint of the human gut microbiota.** *Nature* 2019, **568:**499-504.

22. Miller JR, Zhou P, Mudge J, Gurtowski J, Lee H, Ramaraj T, Walenz BP, Liu J, Stupar RM, Denny R, et al: **Hybrid assembly with long and short reads improves discovery of gene family expansions.** *BMC Genomics* 2017, **18:**541.

23. Amarasinghe SL, Su S, Dong X, Zappia L, Ritchie ME, Gouil Q: **Opportunities and challenges in long-read sequencing data analysis.** *Genome Biol* 2020, **21:**30.

24. Magoc T, Pabinger S, Canzar S, Liu X, Su Q, Puiu D, Tallon LJ, Salzberg SL: **GAGE-B: an evaluation of genome assemblers for bacterial organisms.** *Bioinformatics* 2013, **29:**1718-1725.

25. Zerbino DR, Birney E: **Velvet: algorithms for de novo short read assembly using de Bruijn graphs.** *Genome Res* 2008, **18:**821-829.

26. Wu B, Li M, Liao X, Luo J, Wu F, Pan Y, Wang J: **MEC: Misassembly Error Correction in contigs based on distribution of paired-end reads and statistics of GC-contents.** *IEEE/ACM Trans Comput Biol Bioinform* 2018.

27. Sahlin K, Vezzi F, Nystedt B, Lundeberg J, Arvestad L: **BESST--efficient scaffolding of large fragmented assemblies.** *BMC Bioinformatics* 2014, **15:**281.

28. Mandric I, Zelikovsky A: **ScaffMatch: scaffolding algorithm based on maximum weight matching.** *Bioinformatics* 2015, **31:**2632-2638.

29. van der Walt AJ, van Goethem MW, Ramond JB, Makhalanyane TP, Reva O, Cowan DA: **Assembling metagenomes, one community at a time.** *BMC Genomics* 2017, **18:**521.

30. Smith-Unna R, Boursnell C, Patro R, Hibberd JM, Kelly S: **TransRate: reference-free quality assessment of de novo transcriptome assemblies.** *Genome Res* 2016, **26:**1134-1144.

31. Reppell M, Novembre J: **Using pseudoalignment and base quality to accurately quantify microbial community composition.** *PLoS Comput Biol* 2018, **14:**e1006096.

32. Nurk S, Meleshko D, Korobeynikov A, Pevzner PA: **metaSPAdes: a new versatile metagenomic assembler.** *Genome Res* 2017, **27:**824-834.

33. H L: **Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM.**

preprint at https://arxivorg/abs/13033997?upload=1 2013.

34. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, Genome Project Data Processing S: **The Sequence Alignment/Map format and SAMtools.** *Bioinformatics* 2009, **25:**2078-2079.
35. Breiman L: **Random forests.** *Mach Learn* 2001, **45:**5-32.
36. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al: **Scikit-learn: Machine Learning in Python.** *Journal of Machine Learning Research* 2011, **12:**2825-2830.
37. Liu FT, Ting KM, Zhou Z-HJAToKDfD: **Isolation-based anomaly detection.** 2012, **6:**1-39.
38. Parks DH, Imelfort M, Skennerton CT, Hugenholtz P, Tyson GW: **CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes.** *Genome Res* 2015, **25:**1043-1055.

Figures

Fig. 1 Overall framework of metaMIC. **a** metaMIC extracts four types of features from the alignment of paired end reads to contigs: read coverage, nucleotide variants, mate pair consistency, and *k*-mer abundance consistency. **b** Misassembled contigs are identified by metaMIC based on the four features. **c** metaMIC first identifies the error regions containing misassembly breakpoints, and then recognizes the exact positions of breakpoints and corrects misassemblies by splitting misassembled contigs at breakpoints.

Fig. 2 metaMIC outperforms ALE and DeepMAsED in identifying misassembled contigs in simulated metagenomic datasets. **a-d** The performance of the three tools on the CAMI-medium (M) and high-complexity (H) communities (**a**), *CAMI2-Skin* (**b**), *CAMI2-Gut* (**c**), and simulated virome dataset (*Sim-Virome*) (**d**). **e** The AUPRC scores of metaMIC on test datasets assembled by MEGAHIT or IDBA_UD (Test assembler), where metaMIC were trained on contigs from training datasets assembled by MEGAHIT, IDBA_UD, or jointly by MEGAHIT and IDBA_UD (MEGAHIT+IDBA_UD).

Fig. 3 The performance of metaMIC in localizing misassembly breakpoints on CAMI datasets. **a, b** The distribution of anomaly scores (**a**) and read breakpoint ratios (**b**) of different misassembly types across contigs from CAMI1-Medium. **c** The receiver operation curves by ALE, anomaly scores and read breakpoint ratios when discriminating error regions from error-free regions in CAMI1-Medium, respectively. **d, e** The distribution of error size of misassembly breakpoints recognized by metaMIC on CAMI1-Medium (*Medium*), CAMI1-High (*High*), CAMI2-Skin (*Skin*), CAMI2-Gut (*Gut*) and CAMI2-Oral (*Oral*) (**d**), and different misassembly types in CAMI1-Medium (**e**). **f** The distribution of normalized error size of misassembly breakpoints recognized by metaMIC and ALE on CAMI1-Medium.

Fig. 4 Splitting misassembled contigs at breakpoints improves the downstream binning results over *Sim-Virome* and *CAMI1-Medium* datasets. **a** The number of high-quality bins with low contamination (<5%) of different completeness reconstructed from original and corrected contigs. **b** The distribution of F1 scores for bins reconstructed based on contigs before and after correction, where only those bins whose results change before and after correction were shown for clearness.

Fig. 5 The performance of metaMIC on real metagenomic datasets. **a** The number of bins of different completeness with low contamination (<5%) reconstructed from original and corrected assemblies of '*Ethiopian*' (left) and '*Madagascar*' (right) cohorts. **b** Comparison of F1 scores for reconstructed bins before and after correction of contigs from '*Ethiopian*' (top) and '*Madagascar*' (bottom) cohorts. **c** An example of a predicted misassembled contig "k141_847840" assembled from combined rumen fluid and solid sample. The top plot shows the alignment result of Illumina short-read assembled contig "k141_847840" and PacBio long-read assembled contigs ("contig_982" and "contig_158"), where two regions in the "k141_84780" (1201-6738bp and 6920-8700bp) were aligned to "contig_982" and "contig_158", respectively. The middle figure shows a snapshot of Integrative Genomics Viewer for contig "k141_847840". The bottom plot shows the anomaly score (blue) and read breakpoint ratio (green) across contig "k141_847840".

Tables

Table 1 Performance comparison of metaMIC and MEC on four real datasets from the GAGE-B project.

Species	Correction tool	Misassembled contig length	FN	TP	FP
M. abscessus	raw	1,189,973	20	\	\
	MEC	982,986	14	6	2
	metaMIC	593,741	6	14	2
V. cholerae	raw	597,777	7	\	\
	MEC	597,183	6	1	0
	metaMIC	205,644	3	4	1
R. sphaeroides	raw	135,153	2	\	\
	MEC	64,489	1	1	0
	metaMIC	0	0	2	7
B. cereus	raw	117,830	5	\	\
	MEC	56,086	4	1	0
	metaMIC	28,068	1	4	3

Misassembled contig length denotes the total number of bases in the raw misassembled contigs or the misassembled contigs that cannot be corrected by MEC or metaMIC; True positive (TP) is the number of true misassemblies identified by the error correction tool; False positive (FP) is the number of misassemblies which are actually correct but mistakenly identified as misassemblies; False negative (FN) denotes the number of true misassemblies that are not identified.

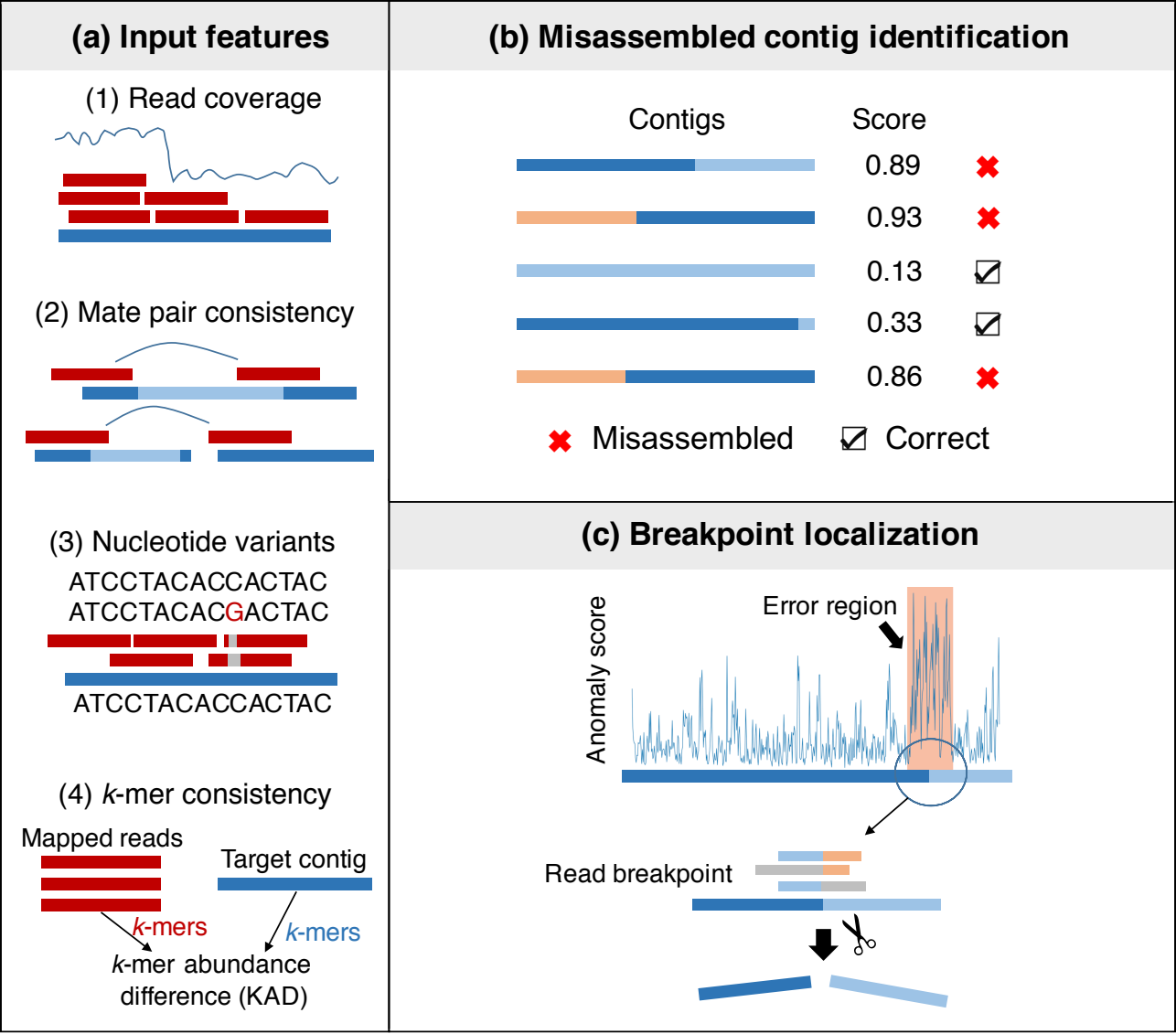
548 **Table 2** Comparison of BESST scaffolding results of contigs before and after correction.

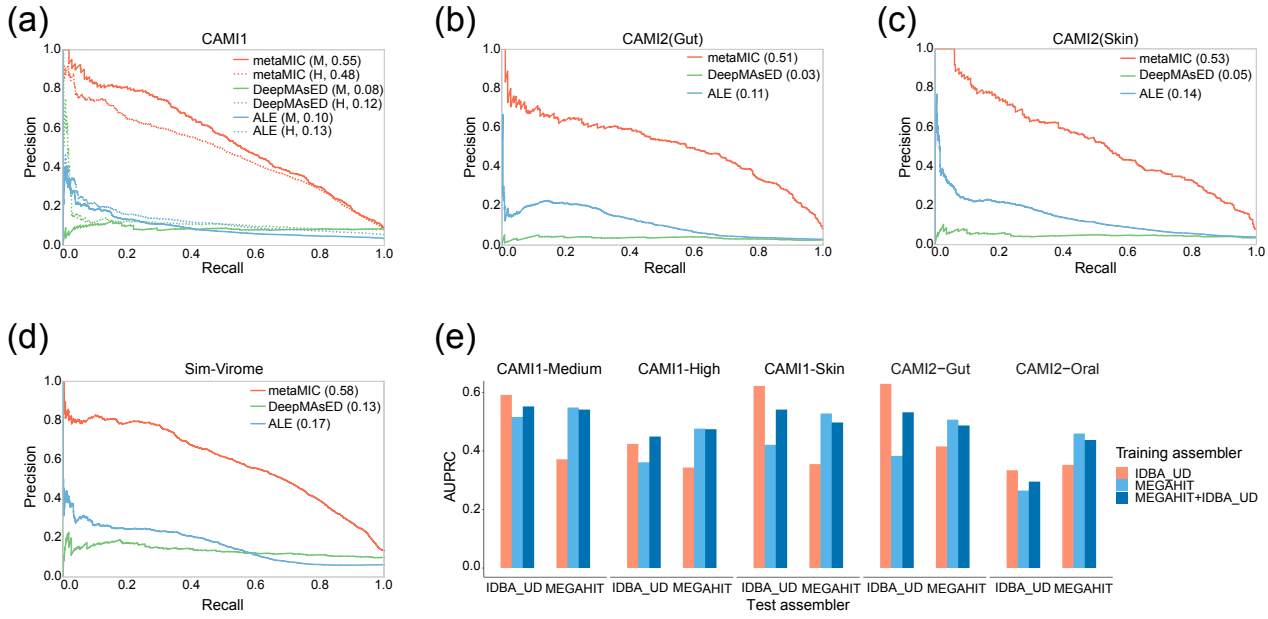
Species	Correction tool	#Contigs	#Total aligned length	Total length(>=0bp)	Total length(>=1000bp)	Misassembled contig length	NA50	#Mis
M.abscessus	raw	262	5,045,398	5,160,404	5,129,190	1,303,084	45,957	27
	MEC	268	5,045,445	5,160,476	5,129,015	982,986	40,129	24
	metaMIC	274	5,045,398	5,160,404	5,129,190	755,186	47,488	17
V.cholerae	raw	201	3,936,390	3,958,533	3,921,645	597,777	43,122	10
	MEC	202	3,935,796	3,958,533	3,921,645	597,183	43,122	9
	metaMIC	205	3,936,390	3,958,533	3,921,645	205,644	43,123	7
R.sphaeroides	raw	231	4,492,687	4,519,491	4,486,060	359,217	75,728	5
	MEC	230	4,492,749	4,519,550	4,486,119	168,646	78,611	4
	metaMIC	232	4,493,107	4,520,061	4,486,630	51,809	78,920	3
B.cereus	raw	141	5,310,597	5,381,347	5,369,165	332,560	104,970	7
	MEC	140	5,310,816	5,381,940	5,369,758	332,001	104,970	7
	metaMIC	140	5,311,395	5,382,650	5,370,789	175,743	104,970	5

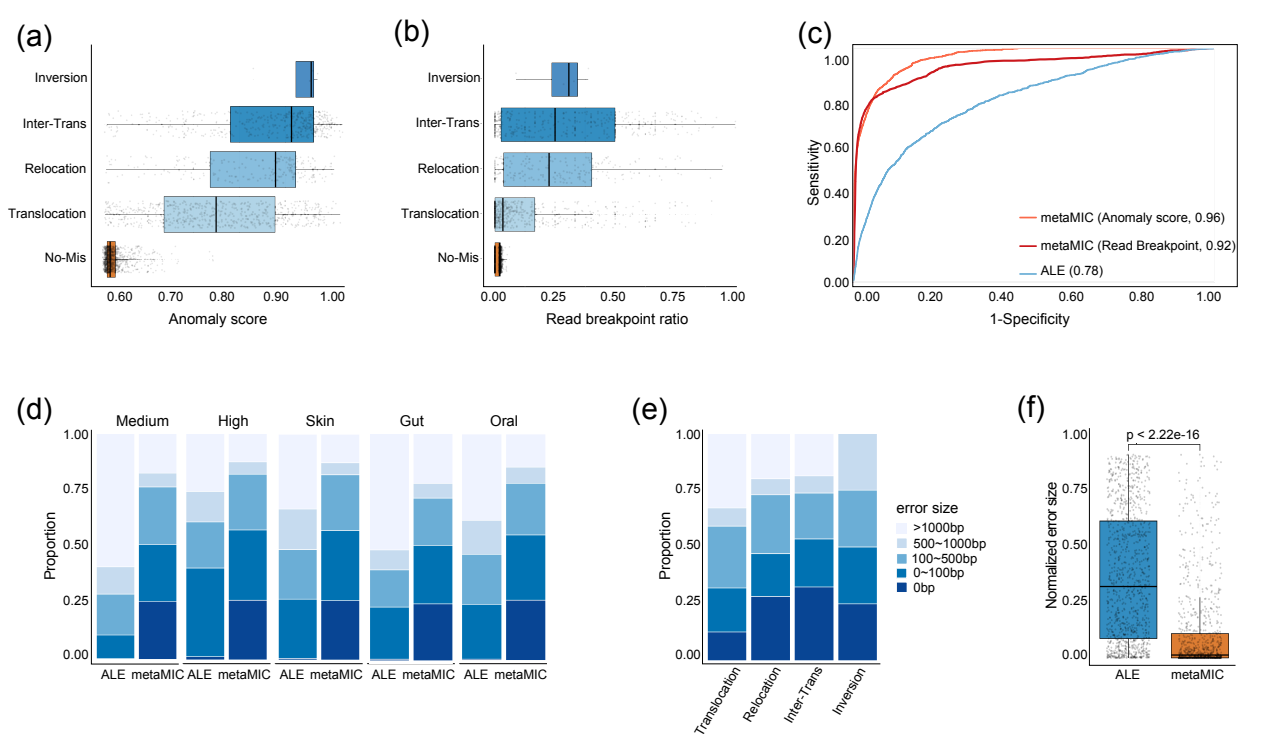
549 #Mis denotes the number of scaffolds that contain misassemblies; Total aligned length denotes the

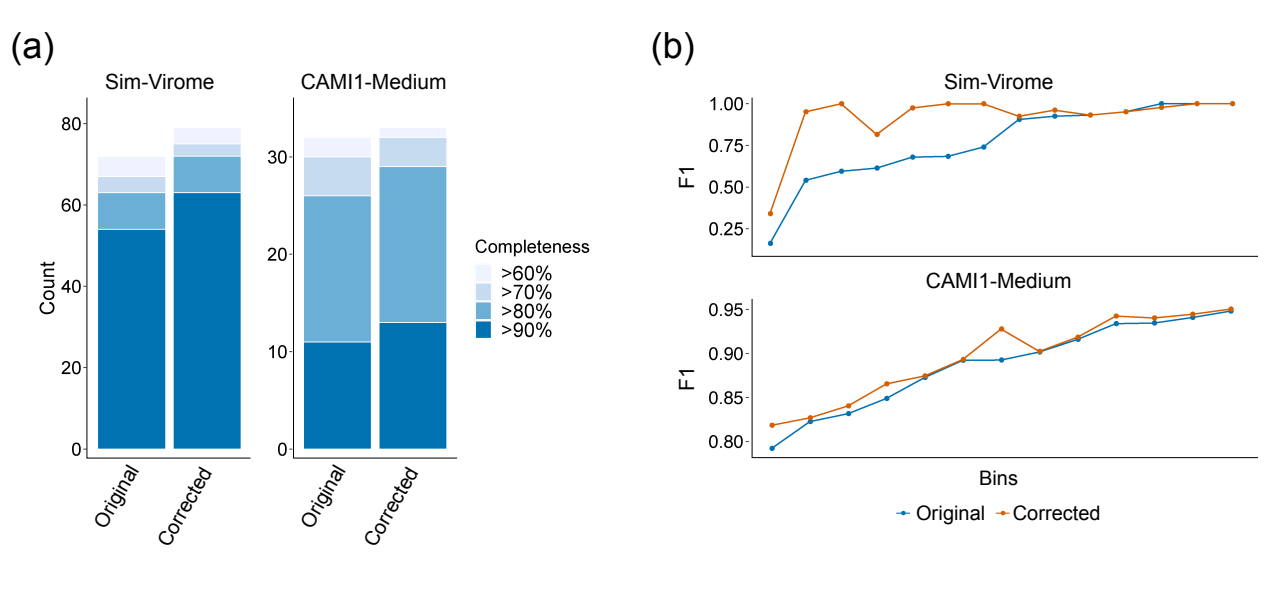
550 length of total number of bases from contigs that can be aligned to the assembly.

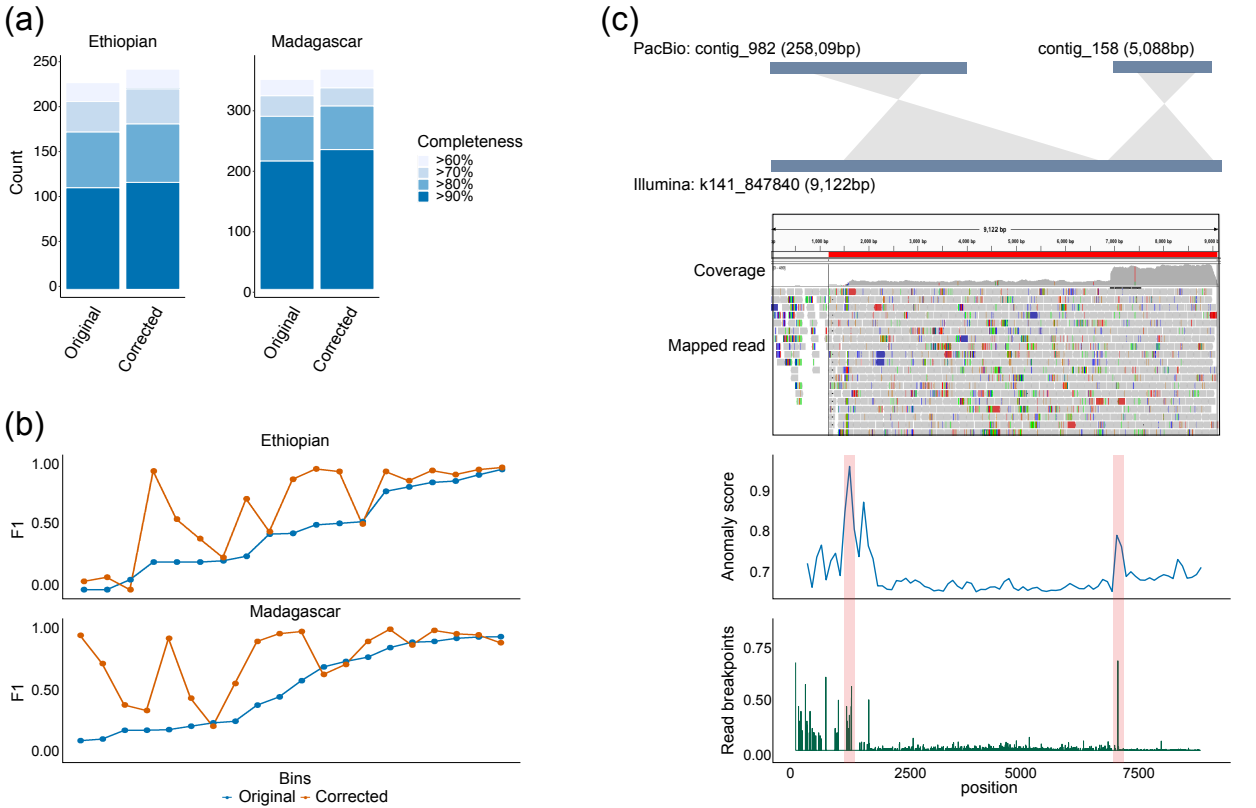
551


(a) Input features
(1) Read coverage
(2) Mate pair consistency
(3) Nucleotide variants
ATCCTACACCACTAC
ATCCTACACGACTAC
ATCCTACACCACTAC
(4) k-mer consistency
Mapped reads
Target contig
k-mers
k-mers
k-mer abundance difference (KAD)
(b) Misassembled contig identification
Contigs
Score
0.89
0.93
0.13
0.33
0.86
Misassembled
Correct
(c) Breakpoint localization
Anomaly score
Error region
Read breakpoint


(a)
CAMI1
metaMIC (M, 0.55)
metaMIC (H, 0.48)
DeepMAsED (M, 0.08)
DeepMAsED (H, 0.12)
ALE (M, 0.10)
ALE (H, 0.13)
Precision
Recall
(b)
CAMI2(Gut)
metaMIC (0.51)
DeepMAsED (0.03)
ALE (0.11)
Precision
Recall
(c)
CAMI2(Skin)
metaMIC (0.53)
DeepMAsED (0.05)
ALE (0.14)
Precision
Recall
(d)
Sim-Virome
metaMIC (0.58)
DeepMAsED (0.13)
ALE (0.17)
Precision
Recall
(e)
CAMI1-Medium
CAMI1-High
CAMI1-Skin
CAMI2−Gut
CAMI2−Oral
AUPRC
IDBA_UD
MEGAHIT
Test assembler
Training assembler
IDBA_UD
MEGAHIT
MEGAHIT+IDBA_UD


(a)
Inversion
Inter-Trans
Relocation
Translocation
No-Mis
0.60
0.70
0.80
0.90
1.00
Anomaly score
(b)
Inversion
Inter-Trans
Relocation
Translocation
No-Mis
0.00
0.25
0.50
0.75
1.00
Read breakpoint ratio
(c)
Sensitivity
1-Specificity
0.00
0.20
0.40
0.60
0.80
1.00
metaMIC (Anomaly score, 0.96)
metaMIC (Read Breakpoint, 0.92)
ALE (0.78)
(d)
Medium
High
Skin
Gut
Oral
Proportion
0.00
0.25
0.50
0.75
1.00
ALE
metaMIC
(e)
Proportion
error size
>1000bp
500~1000bp
100~500bp
0~100bp
0bp
Translocation
Relocation
Inter-Trans
Inversion
(f)
p < 2.22e-16
Normalized error size
0.00
0.25
0.50
0.75
1.00
ALE
metaMIC


(a)
Sim-Virome
CAMI1-Medium
Count
80
60
40
20
0
30
20
10
0
Original
Corrected
Completeness
>60%
>70%
>80%
>90%
(b)
Sim-Virome
F1
1.00
0.75
0.50
0.25
CAMI1-Medium
0.95
0.90
0.85
0.80
Bins
Original
Corrected


(a)
Ethiopian
Madagascar
Count
Completeness
>60%
>70%
>80%
>90%
Original
Corrected
(b)
F1
Bins
Original
Corrected
(c)
PacBio: contig_982 (258,09bp)
contig_158 (5,088bp)
Illumina: k141_847840 (9,122bp)
Coverage
Mapped read
Anomaly score
Read breakpoints
position