Briefings in Bioinformatics, 2017, 1–11

doi: 10.1093/bib/bbx098
Paper

Metagenomic assembly through the lens of validation: recent advances in assessing and improving the quality of genomes assembled from metagenomes

Nathan D. Olson*, Todd J. Treangen*, Christopher M. Hill, Victoria Cepeda-Espinoza, Jay Ghurye, Sergey Koren and Mihai Pop

Corresponding author. Mihai Pop, Center for Bioinformatics and Computational Biology, 8314 Paint Branch Dr. Rm. 3120F, College Park, MD, 20742, USA. E-mail: mpop@umiacs.umd.edu
*These authors contributed equally to this work.

Abstract

Metagenomic samples are snapshots of complex ecosystems at work. They comprise hundreds of known and unknown species, contain multiple strain variants and vary greatly within and across environments. Many microbes found in microbial communities are not easily grown in culture making their DNA sequence our only clue into their evolutionary history and biological function. Metagenomic assembly is a computational process aimed at reconstructing genes and genomes from metagenomic mixtures. Current methods have made significant strides in reconstructing DNA segments comprising operons, tandem gene arrays and syntenic blocks. Shorter, higher-throughput sequencing technologies have become the *de facto* standard in the field. Sequencers are now able to generate billions of short reads in only a few days. Multiple metagenomic assembly strategies, pipelines and assemblers have appeared in recent years. Owing to the inherent complexity of metagenome assembly, regardless of the assembly algorithm and sequencing method, metagenome assemblies contain errors. Recent developments in assembly validation tools have played a pivotal role in improving metagenomics assemblers. Here, we survey recent progress in the field of metagenomic assembly, provide an overview of key approaches for genomic and metagenomic assembly validation and demonstrate the insights that can be derived from assemblies through the use of assembly validation strategies. We also discuss the potential for impact of long-read technologies in metagenomics. We conclude with a discussion of future challenges and opportunities in the field of metagenomic assembly and validation.

Nathan D. Olson is a PhD student at the University of Maryland and researcher at the National Institute of Standards and Technology working on methods for evaluating metagenomic bioinformatic tools.

Todd J. Treangen received his PhD in Computer Science from the Technical University of Catalonia, Barcelona Spain. His research interests include multiple genome alignment and metagenomic assembly. Currently, he is an Assistant Research Scientist at the Center for Bioinformatics and Computational Biology, University of Maryland College Park.

Christopher M. Hill received his PhD in Computer Science from University of Maryland, College Park, where he focused on developing algorithms for assembling and comparing single genome and metagenomic assemblies. Currently, he is a software engineer at Google Inc.

Victoria Cepeda-Espinoza is a PhD student in the Department of Computer science at the University of Maryland, College Park. Her research interests include developing algorithms for metagenomic assembly.

Jay Ghurye is a PhD student in the Department of Computer science at the University of Maryland, College Park. His research interests include developing algorithms for metagenomic assembly.

Sergey Koren received his PhD in Computer Science from the University of Maryland. His research interests include single-molecule sequence assembly and analysis. He is currently a Staff Scientist in the Genome Informatics Section at the National Human Genome Research Institute.

Mihai Pop received his PhD degree in Computer Science from Johns Hopkins University. His research interests include sequence analysis algorithms and metagenomics. Currently, he is a Professor in the Department of Computer Science and the Center for Bioinformatics and Computational Biology, and the Interim Director of the Institute for Advanced Computer Studies at the University of Maryland, College Park.

Submitted: 2 May 2017; **Received (in revised form):** 13 July 2017

Key words: metagenomics; microbiome; metagenomic assembly; assembly validation; variant discovery

Introduction

Shotgun sequencing of microbial communities, metagenomics, has emerged as a key tool for investigating the composition, evolutionary history and function of communities comprising previously uncultured and unsequenced organisms. Assembling metagenomic sequencing data can provide a more complete picture of a microbial community compared with performing analyses directly on the reads [1–4]. However, assembly and metagenomic assembly are complex computational tasks. The complexity of the assembly problem stems from the DNA segments repeated within a same organism, or shared between distinct organisms. Intragenomic repeats have long been recognized as a challenge in assembly of isolate genomes [5], while metagenomic assembly is further complicated by the combination of intragenomic and intergenomic repeats. It has been shown that assembly complexity is directly tied to the ratio of the sequencing read length and the length of repeats [6]. While intergenomic repeats are generally small (usually ∼<10 000 bp in bacteria [7, 8]), intragenomic repeats can be nearly the entire chromosomes for closely related strains. To better explain this perhaps un-intuitive statement—just one gene (e.g., a virulence gene) may distinguish between two closely-related bacterial genomes in a community, thus close to the entire genome can be viewed as an intergenomic repeat.

Sequencing strategies for metagenomics are currently dominated by short, high-throughput sequencing technologies, such as the Illumina NextSeq and HiSeq. These technologies can produce billions of highly accurate 100–300 bp reads within a few days and are cost-effective for most large-scale microbiome research projects. Metagenome sequence assembly algorithms have been largely based on a de Bruijn graph (DBG) paradigm, which is effective for accurate [9] and efficient assembly of large metagenomic data sets [10]. New advances in sequencing methods, such as single-molecule sequencing, synthetic long reads and Hi-C along with new assembly and scaffolding algorithms have the potential to significantly improve the contiguity and quality of metagenome assemblies, and are an emerging area of research interest. To date, however, the use of such technologies in metagenomic settings has been limited because of the complex sample processing requirements and cost.

Owing to the complexity of the metagenome assembly problem, regardless of the assembly algorithm used or sequencing method, metagenome assemblies are incomplete and contain errors. Methods for evaluating the quality and completeness of assemblies are critical for informing downstream analyses of the assembled data, and for allowing researchers to compare different tools that could be used for assembly.

Assembly validation methods fall into two broad categories: reference-based and *de novo*. Reference-based validation methods compare the assembly with a database containing previously assembled genes or genomes [11, 12]. They assess as errors any differences identified between the assembled data and the reference collection. In contrast, *de novo* methods rely on features of the assembled data itself, seeking to identify internal inconsistencies indicative of potential assembly errors. Reference-based methods are particularly effective in benchmarking experiments attempting to reconstruct communities with known composition; however, these methods have limited effectiveness in real data sets. For example, metagenomic segments originating from a genome for which no reference sequence is available cannot be verified through a reference-based approach. It is also difficult to determine whether differences between an assembled contig are errors or true differences between the reference sequence and its relative within the metagenomic mixture.

Current methods for metagenome assembly

The (meta)genome assembly problem can be formulated as a graph traversal problem, finding a path through a complex graph satisfying the constraints imposed by the data provided to the assembler [13]. Metagenome assembly is accomplished *de novo* by reconstructing genomes directly from the read data [13]. Despite the development of dozens of implementations for *de novo* assembly, algorithmic challenges posed by repeats remain prohibitive (Figure 1).

The problem of reconstructing a mixture of genomes is further complicated by uneven and unknown representation of the different organisms within a metagenomic mixture. Owing to uneven sequencing coverage within a metagenome, coverage heuristics used for isolate genome assembly cannot be readily used to accurately disentangle repetitive sequence in metagenomes [14]. This problem is further aggravated by the presence of intergenomic and intragenomic repeats (Figure 1). Effectively, one can view the task of the assembler to be not just to reconstruct one path through a graph, but a multitude of paths that come together and split apart at different places. This problem is not exclusive to metagenomic assembly—polyploid plant and animal genomes cause similar issues for isolate genome assembly. Prior work in polyploid genomes has provided methods for estimating ploidy based on sequencing depth, *k*-mer distribution and genotype heterogeneity [15]. Even so, despite initial attempts, algorithms developed for single-genome assembly have

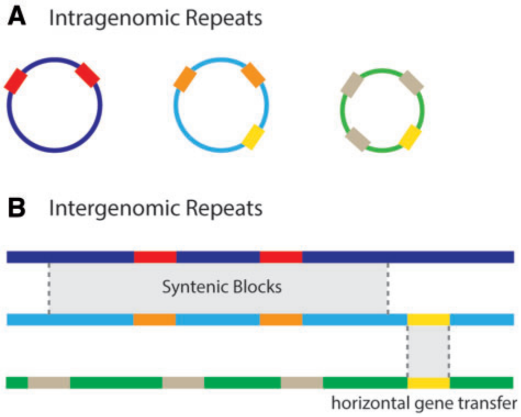


Figure 1. The challenge of repeats in metagenomes. Three genomes are used to depict intragenomic (**A**) and intergenomic (**B**) repeats. The dark blue and light blue genomes represent two closely related strains and the green genome an unrelated strain. Within the genomes, the red, orange and tan blocks represent inparalogs. The yellow blocks represent a horizontal gene transfer event between the light blue and green genomes. In traditional assembly, any reads longer than the inparalog blocks (red; orange) would be sufficient to fully resolve the genome. In metagenomic assembly, reads longer than the full syntenic block (gray) would be necessary.

not been successfully applied directly to metagenomics data. Instead, several approaches have been developed that explicitly consider the specific characteristics of metagenomic data. Below, we describe a few of these approaches. The main approaches described here generally try to 'hide' the complexity imposed by metagenomic data. We will later discuss in more detail approaches that specifically try to identify strain variants in the data. Before doing so, however, we provide a short overview of the DBG approach for assembly and the impact of the parameters of this approach on the assembly process.

Briefly, the reads are converted to a graph as follows. Each read is decomposed into overlapping segments of equal length *k*, usually termed *k*-mers. The *k*-mers become the nodes of the graph, and the edges connect nodes with $k-1$ matching bases. It is also possible to define the DBG in a node-centric manner (edges indicate $k-1$ matching bases between nodes; see [16] for more details). In a metagenomic context, one looks for multiple paths through the graph that collectively 'explain' all the edges. While an exact algorithm exists that can solve the traversal problem efficiently, it can only find one out of the many possible traversals of the graph that are consistent with the set of reads, reconstructing a possibly incorrect sequence representing a rearrangement of the genome(s). The above formulation allows traversals, which are not necessarily consistent with the input sequences, and adding as a constraint the reads themselves leads to the 'Eulerian superpaths' [13] formulation. In general, there are often multiple valid traversals of the DBG and identifying the correct one is the source of computational complexity in the assembler [6]. Most assemblers rely on heuristics to incorporate further information in the reconstruction process to bias the reconstruction toward the correct sequence. When ambiguities cannot be resolved, the assemblers break the traversal, leading to fragmented reconstructions of the original genome(s).

Several factors impact the performance of DBG assemblers: (i) sequencing errors, (ii) repeats, (iii) the presence of strain variants and (iv) the depth of sequencing coverage. The interplay between these factors drives the choice of optimal *k*-mer size for a specific application as well as the ultimate performance of an assembler. Sequencing errors create 'false' *k*-mers, thereby increasing the complexity of the graph and making it more difficult to identify an unambiguous reconstruction of a sequence. Every error impacts at most *k* different *k*-mers; thus, the impact of sequencing errors increases with the size of *k*. The de Bruijn formulation above assumes perfect data. In practice, sequencing errors introduce false *k*-mers in the graph, increasing the size of the graph and adding ambiguity in the reconstruction. As a result, assemblers often include a 'correction' step or assume precorrected data as input. Initial de Bruijn assemblers used spectral correction [13], which attempts to make a minimum number of changes in a sequence to make it consistent with 'correct' or 'solid' *k*-mers [17, 18]. The correction strategies use a fixed *k*-mer count threshold to define 'correct' *k*-mers, strategy that is insufficient in metagenomic data sets with varying coverage levels. Recent approaches to correction have been proposed, which can correct data without assuming uniform coverage [10, 19, 20].

Repeats create additional edges in the graph, increasing the number of possible traversals. This creates ambiguity in the reconstruction of the genome, as a larger possible space of solutions must be explored [6]. Without further information, an assembler can either choose one of the branches at random, possibly leading to assembly errors, or simply decide to break the assembly, leading to fragmented results. The longer the size of *k*, the fewer nodes in the graph are repetitive, and thus, the easier it is to reconstruct large segments of a genome. Strain variants create a similar challenge as sequencing errors, and in highly polymorphic samples, the assembly result will likely be fragmented [21]. Finally, the depth of coverage impacts the connectivity of the assembly graph. A path stretching from a read to the next until it covers an entire genome can only be found if adjacent reads share *k*-mers. At low depths of coverage, the adjacent reads are only expected to overlap by a small extent, and as a result, the assembly is only possible for small values of *k*.

To summarize the points made above, large values of *k* reduce the complexity of the graph and impact of repeats, but using such values requires longer sequences (longer than the *k*-mer size) and higher depth of coverage, and leads to an increased impact of sequencing errors (each error impacts *k* different *k*-mers). Assuming uniform error and random sequencing, it is possible to compute the expected surviving coverage for a given *k*-mer size and input coverage [22]. These trade-offs represent a key component of the algorithmic choices made by assembly software and also guide the empirical choices made by users of assembly tools.

In the following section, we highlight three published algorithms developed specifically for metagenome assembly that perform well in a recent review [23].

IDBA-UD

IDBA-UD [24] is part of the IDBA (Iterative De Bruijn Graph *De Novo* Assembler) [25] suite of assemblers. IDBA assemblers use multiple *k*-mer sizes to address the trade-offs described above. It iterates through a range of *k*-mer values in a stepwise fashion to improve the DBG and resulting assembly. Sequencing errors are corrected at each iteration, reducing the impact of sequencing errors. In this way, the assembly graph becomes more and more resolved with increasing *k*-mer size in each iteration step, leading to a more contiguous assembly result.

MEGAHIT

MEGAHIT [10] relies on the same multiple *k*-mer strategy as the [10] IDBA assemblers [25]. MEGAHIT is currently the most efficient *de novo* assembler largely because of its use of efficient data structures for storing the DBG. Memory requirements are reduced by using a new data structure, a succinct DBG [26]. Memory is also reduced by eliminating *k*-mers below a defined frequency threshold from the graph. This approach minimizes the negative impact of sequencing errors on the assembly. To retain *k*-mers from low-abundance organisms, distinguishing them from errors, MEGAHIT reconsiders discarded *k*-mers in low-coverage regions of the assembly graph.

metaSPAdes

MetaSPAdes [9] is a metagenomic-specific version of the SPAdes assembler [27]. A main innovation in these assemblers is the use of paired-end information during the assembly process rather than afterward [28]. This information is incorporated in the graph by using a pair of k-mers separated by an estimated distance. Similar to IDBA-UD and MEGAHIT, SPAdes uses an iterative multiple k-mer approach. However, SPAdes uses the complete read information together with the preassembled contigs at every step. Originally, SPAdes was designed to address two major issues of single-cell sequencing data [27], the uneven read coverage and chimeric sequences, issues that are also germane to metagenomic assembly. In addition, metaSPAdes [9] was extended to handle strain variation. Micro-variations between highly similar 'strain-contigs' are combined to form high-quality consensus sequences, aiming at the best possible representation of each species instead of every strain variant.

Metagenome scaffolding

To improve the continuity of fragmented assemblies, orthogonal information, which is not used in the assembly process, is used to orient and order contigs with respect to each other. The linkage information provides a measure of confidence about the proximity of any two contigs on the genome. Different kinds of linkage information such as optical maps [29], mate pairs [30–33], fosmid clones [34], fosmid clone dilution pool sequencing [35], linked read sequencing [36], synthetic long reads [37] and Hi-C [38–40] have been explored to improve genome assembly quality. Paired-end sequences have a known size distribution, which can give an estimate of the distance between two contigs. Hi-C makes use of the 3D structure of the chromosomes inside a cell nucleus, thereby inferring genome-scale contact information. Development of algorithms, which use one or more types of orthogonal linkage information to get high-quality assemblies, is an emerging area of interest. However, the applicability of these methods in metagenomics has been limited because of difficult and expensive sample processing protocols. Because of this, the information used for scaffolding metagenomes has been limited to paired-ends.

Strain resolution and variant detection

Unlike isolate genome sequencing, the analysis of microbial communities provides valuable information about the strain structure of mixtures of closely related organisms, making it possible to study how the strain composition changes across time or in response to environmental changes [41–45]. This power has been recognized since the early days of the field [42], and software packages have been developed to help scientists discover, characterize and quantify strain-level differences between microbes. A first package in the field, Strainer [46], allowed researchers to manually inspect metagenomic assemblies to identify single-nucleotide variants. The Bambus 2 [47] scaffolding package was the first tool to provide the ability to automatically detect structural variants within metagenomic assemblies. This approach was later extended in Marygold [48] through the use of SPQR trees [49] to allow the efficient discovery of more types of structural variants. Most recently Anvi'o [50] provides a data analytics and visualization environment for comparing strain variants across multiple data sets. Using available infant gut samples [43], Anvi'o allowed the identification of systematic emergence of nucleotide variation in an abundant draft genome bin [50]. Note that the problem of strain resolution in metagenomic data bears strong similarities to the reconstruction of transcript splicing structure in RNA sequencing data [51]. Owing to the much shorter extent of eukaryotic transcripts, and the fact that transcript graphs can be assumed to lack cycles (which is not true in metagenomes), the approaches developed in this field cannot be effectively applied to metagenomic data.

Future strategies and approaches

High-quality assembly of bacterial genomes has undergone a renaissance with the advent of single-molecule sequences [52, 53], such as the PacBio RSII/Sequel [54] and Oxford Nanopore MinION [55]. An alternate technology is TruSeq Synthetic Long Reads (TSLRs, previously known as Moleculo), which relies on barcodes and pooling to reconstruct long sequences [37, 56, 57]. While not yet used for resolving microbial genomes [58] because of difficulty in assembling variable number tandem repeats, the TSLR sequences have high accuracy, allowing strain-level resolution in metagenomes [59]. Single-molecule sequencing is continuing to mature, increasing in both throughput and quality. As the instruments have increased in throughput, they have been applied to assemble eukaryotic genomes [60–67] with improved assembly algorithms [62, 68–70].

In addition to long-read sequencing, novel scaffolding approaches have also had a significant impact on genome assembly. The combination of Hi-C scaffolding and long-read assembly has been particularly powerful in combination, generating chromosome-scale scaffolds [60, 71].

Recent studies have begun to highlight the application of long reads to metagenomics [72] across a broad range of applications such as: gut microbiome [73], coculture communities [74] and the skin microbiome [75]. However, long-read sequencing has not gained widespread adoption because of three main factors: cost, DNA quality requirements and complexity of DNA preparation. A prerequisite for the effectiveness of both synthetic long-read and single-molecule long-read sequencing is that the DNA fragments provided to the sequencing instrument are sufficiently long. Current DNA extraction procedures, in addition to lysing the cell, also shear the DNA, thereby limiting sequencing read length [76]. This is especially true in the case of gram-positive organisms that are difficult to lyse. While protocols can be effectively optimized on a per-organism basis, this is not true for complex mixtures. Novel DNA extraction methods using agarose plugs, like those used for extracting DNA for optical mapping [77, 78], or enzymatic lysis using cocktails of enzymes [79], may result in the extraction of longer DNA fragments. Long-range linking technologies, such as Hi-C, have an advantage over long-read sequencing, as these technologies do not require high molecular weight DNA to be generated after cross-linking and cell lysis.

Current methods and strategies for metagenome assembly validation

The validation of genome assemblies has been an active area of interest since the development of the first genome assemblers in the late 1970s [80]. Below, we describe several of the strategies and corresponding used in this context (Table 1).

Some of the most intuitive metrics relate to assembly contiguity. Measures such as the number of contigs and average or maximum contig sizes, attempt to assess how far the assembly is from the ideal objective of one contig per chromosome. As most assemblies comprise many small contigs, usually because of sequencing errors or other artifacts, these metrics can be misleading. A more robust measure is the N50 size, defined as the minimum contig length in the set of contigs that comprise over half the assembly (a weighted median contig size). Other metrics refer to the information contained in contigs, such as the number of open reading frames (ORFs, a proxy for genes) or their density (ORFs/Mb). As genes are used to address biological questions, a greater number or density of ORFs result in more information available for testing biological hypotheses. Contig length statistics do not incorporate any correctness information and can be 'fooled' by accepting errors (a single long contig can be constructed by concatenating all the reads in an arbitrary

Table 1. Metrics to evaluate assembly quality

Assembly metric		Description	Features			
			Reference-based	Reference-free	Measures errors	Measures errors + variation
Contiguity-based metrics	Number of contigs	Total number of assembled contigs reported by each assembler	√	√		
	Assembly size at 1 Mb	Represents the size of the largest contig C such that the sum of all contigs larger than C exceeds 1 Mb		√		
	Contig number at 1 Mb	Represents the number of contigs required to exceed 1 Mb		√		
	Complete genes	Represents the median number of complete genes per sample		√		
	Complete marker genes	Indicates the median number of fully reconstructed marker genes per sample	√			
Reference-based metrics	Genome recovery (%)	Median percentage of each truth genome that is recovered	√			√
	Total aligned length	Sum of the length of contigs aligned to the truth genomes	√			√
	Total unaligned length	Sum of the length of unaligned contigs	√			√
	NGAx	The length of the contig that covers at least half the reference genome. Contigs are broken at mis-assembly events and removing all unaligned bases	√			√
Consistency- based metrics	Depth of coverage	Statistical comparison of global versus local coverage, as signature of compressed/expanded repeats, chimeric contigs		√	√	
	Consensus	Concordance of consensus to read pileup		√	√	
	Split-read mapping	Single reads with partial alignments		√	√	
	Insert size consistency	Concordance of insert size (expanded/collapsed)		√	√	

Note: Most commonly used metrics to evaluate metagenomics assembly quality, including contiguity-based, reference-based and consistency-based metrics. Assembly metric column contains commonly used metrics; description column briefly describes the metric; and features column indicates four characteristics of these metrics: reference-based (reference genome required), reference-free (reference genome not required), measures errors, measures errors + variation (biased by real differences in reference genome).

order). In contrast, ORF-based statistics capture error, as they would disrupt ORFs.

ORF/gene information can also be used to evaluate assembly completeness. Several genes (called marker genes) have been found in all known bacterial genomes and can thus be assumed to exist in a newly assembled sequence. An assembly where some of these genes are missing can be assumed incomplete. A recently developed software package, CheckM [12], relies on marker genes that are specific to a genome-based lineage within a reference tree. CheckM also provides tools for identifying sets of contigs that can be combined to reconstruct individual organisms, based on marker set compatibility, similarity in genomic characteristics and proximity within a reference genome tree.

An alternative approach for validating assemblies assesses the fit of the assembly with a model of the sequencing process. Assembly likelihood estimators have been developed and used to evaluate single-genome assemblies [81, 82], as well as metagenome assemblies [83, 84]. While these metrics do not provide a global confidence value, they can be used to compare and rank different genome assemblies, allowing one to automatically optimize the assembly parameters by iteratively trying different options and selecting the parameter set that maximizes the likelihood [83]. Such approaches are largely used as ‘holistic’ validation strategies, assessing the assembly as a whole rather than identifying specific errors. However, they can be adapted to also highlight regions of the assembly where errors may occur [84].

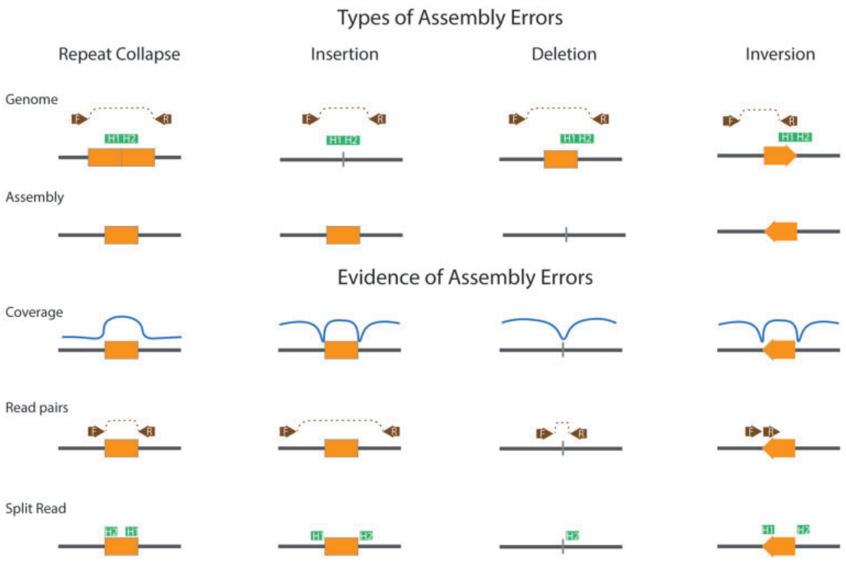
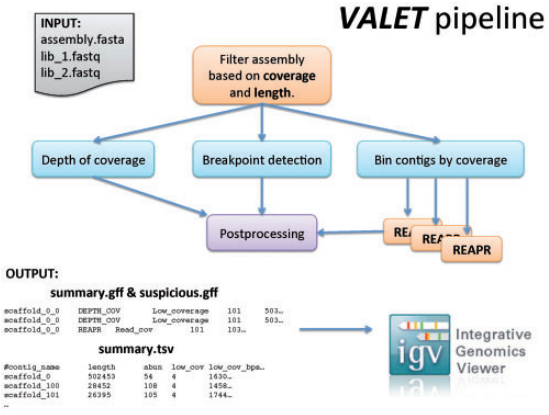


Figure 2. Metagenome assembly error signatures. There are four primary types of assembly errors: repeat collapse, insertions, deletions and inversions. These assembly errors can be identified by mapping reads to the assembly and evaluating the coverage (solid curve), distance between read pairs (boxes labeled H) and split read mapping data (boxes labeled H). Increase in coverage indicates repeat collapse, whereas drops in coverage indicate break points for insertions, deletions and inversions. Shorter than expected distance between read pairs indicates potential repeat collapse or deletion, whereas increase in distance between read pairs indicates a potential insertion. Inconsistency in read pair direction can indicate an inversion. Finally, split-read mapping data, obtained by independently aligning the first and last third of a read can be used in a similar manner to read pair information to identify assembly errors [85].

It is important to note that exactly computing the likelihood of an assembly given a model of the sequencing process can be expensive, simple heuristics are remarkably effective and, in fact, the likelihood metrics are tightly related to the number of reads and/or paired-ends that can be correctly aligned to the assembly. Essentially, the more information that is 'explained' by the assembly, the more likely it is that the assembly is correct.

Characterizing assembly errors

The approaches described so far only implicitly take errors into account. It is often important to determine exactly where errors were introduced in the assembly, either to correct these mistakes or to ensure that the errors do not influence the results of downstream analyses. Figure 2 highlights the four primary types of assembly errors: repeat collapse, insertions, deletions and inversions. These assembly errors can be identified by mapping reads to the assembly and evaluating the coverage, evaluating the distance between read pairs and split read mapping data. Increases in coverage indicate under-collapsed repeats, while drops in coverage or coverage gaps can indicate break points because of insertions, deletions and inversions. There are two primary approaches for detecting these assembly errors: (i) reference-based and (ii) consistency-based (Table 1). In reference-based assembly error detection, assembly errors are identified by comparing the assembly to one or more reference genomes. In consistency-based methods, errors are identified by aligning the sequencing reads to the assembly and identifying regions, where the mappings are inconsistent with the assembly.



```
scaffold_0_0    DEPTH_COV       Low_coverage    101     503…
scaffold_0_0    DEPTH_COV       Low_coverage    101     503…
scaffold_0_0    REAPR   Read_cov        101     103…
```

```
#contig_name    length      abun  low_cov low_cov_bps…
scaffold_0      502453      54    4       1630…
scaffold_100    28452       108   4       1458…
scaffold_101    26395       105   4       1744…
…
```

Figure 3. Overview of the VALET pipeline.

Reference-based

MetaQuast. MetaQUAST [11] is a reference-based method that identifies mis-assemblies and structural variants in an assembly relative to reference genomes. MetaQUAST is a modification of QUAST [86], an isolate genome assembly validation tool that computes alignments of assembled contigs to a single reference genome. For data sets with known reference genomes, metaQUAST uses the user-provided reference sequences to evaluate the assembly. For the data sets where genomes in the sample are not known, metaQUAST identifies appropriate reference sequences using a 16 S ribosomal RNA database. Additionally, metaQUAST applies a structural variant finding algorithm to distinguish between structural variants and true assembly errors.

Consistency-based

Within isolate genomes, errors introduce inconsistencies in the placement of reads within the assembly, leading to several signatures that can be detected computationally (Figure 2). In isolate genomes, several assumptions are usually made. First, the ideal sequencing process can broadly be assumed to be uniform [87], i.e. the DNA fragments are equally likely to start at any position in the genome. Strong deviations from the assumption of uniformity frequently correspond to assembly mistakes. Second, the reads agree with the assembled sequence except for potential random sequencing errors. Third, in the case of paired-end data, the distance between the paired reads is consistent with the fragment sizes generated during the sequencing process. Amosvalidate [88] is a *de novo* pipeline for detecting mis-assemblies that checks all these constraints and reports regions in the assembly characterized by a sufficient deviation from the assumptions outlined above. FRC^{bam} is an approach based on amosvalidate that introduced the concept of feature-response curves (FRCs), which track assembly error across assembled base pairs [89, 90]. REAPR [91] focuses on paired-end constraints, and identifies regions where the distance between the paired-ends is consistently stretched or shrunk, or where the depth of coverage is unusual. Pilon [85] relies on both paired-end information (similar to REAPR) and also on single-base changes and fragmented alignments (called soft-clipping) to identify and correct assembly errors.

None of the consistency-based tools described above are effective in a metagenomic setting, in no small part, as the underlying assumptions are incorrect in this context. Strain variants within the community can be mistaken for errors, and the difference in abundance between the organisms in a mixture makes it difficult to assess what deviations from expectation are 'unusual'.

VALET [92] (Figure 3, http://github.com/marbl/VALET) is a *de novo* pipeline for detecting all types of mis-assemblies in metagenomic data sets (Figure 2). VALET primarily adapts the approaches developed in the context of isolate genomes that we described above. To avoid false positives and false negatives because of uneven depth of coverage, VALET bins contig by coverage before applying these methods.

Possible break points in the assembly are found by examining regions, where a large number of parts of the reads are unable to align. To identify break points, VALET uses the first and last third of each unaligned read, called sister reads. The sister reads are aligned independently to the reference genome, and then regions where the sister reads align to nonadjacent segments of the genome are flagged as mis-assemblies.

In practice, most mis-assembly signatures have high false-positive rates. This false-positive rate can be reduced by focusing on just the regions where multiple signatures agree. Any window of the assembly (2000 bp in length by default) that contains multiple mis-assembly signatures is marked as suspicious by VALET. The flagged and suspicious regions are stored in a BED file, which allows users to visualize the mis-assemblies using genomic viewers, such as IGV [93]. Excluding from the analysis regions of the assembly where just one type of inconsistency is detected may lead to false negatives. It is important for the user to be aware of this trade-off and use the set of signatures that is most appropriate for their application. VALET provides several visual representations of assembly quality including an FRC plot, which highlights the trade-off between contiguity and accuracy.

Exemplar validation of metagenomic data sets

Reference-based and *de novo* evaluation of an HMP data set

To evaluate state-of-the-art metagenomic *de novo* assembly software (IDBA-UD; MEGAHIT; metaSPAdes) on a real data set, we compared the accumulated errors versus cumulative assembly length for multiple assemblies on a Human Microbiome Project (HMP) stool sample (SRS016203) (Figure 4A–C), using FRCs [90]. The results show that different validation metrics provide a different picture of the relative accuracy of the different approaches. In the reference-based MetaQUAST results, MEGAHIT outperforms metaSPAdes and IDBA with metaSPAdes containing the most error (mostly within the shortest contigs). The *de novo* validation based on coverage only (Figure 4B) favors metaSPAdes, while the validation based on break point events favors MEGAHIT. MEGAHIT has fewer structural errors compared with metaSPAdes, especially in the largest contigs, while metaSPAdes has fewer under-collapsed/over-collapsed repeats when compared with MEGAHIT. Taken together, these validation results highlight the need for 'use-case' specific metagenomic assembly pipelines. Depending on the application, different types of errors have varying impacts on the final results. For example, coverage errors make it difficult to estimate relative abundances, thereby possibly confounding statistical associations based on the assembled data. On the other hand, errors with respect to a reference database are most relevant when metagenomic assembly is applied to clinical data sets, setting where most pathogens can be assumed to exist in reference collections.

Reference-based evaluation of long-read assemblies

We next evaluated the promise of long-read assembly using an available HMP synthetic data set sequenced with both PacBio [94] and TSLR [59] assembled with Canu [68]. We compared the results with assemblies of the same data set using short-read data. The full PacBio-only data set (median coverage 192-fold) generates the most complete representation of the data set, reconstructing 67/83 Mb of the data set with several genomes and plasmids in complete circular contigs (Figure 5). This assembly also has a low rate of mis-assembly per megabase (0.23/100 kb) and the lowest mismatch rate (1.87/100 kb). It also has a low insertion and deletion (indel) rate (3.83/100 kb), second only to TSLR sequencing. As with clonal bacterial assemblies, despite the high raw error rate of individual sequences, the consensus assembly has high accuracy exceeding that of short-read data sets because of the signal-based polishing of the consensus sequence [96]. A one-tenth subset PacBio-only assembly (median coverage 19.5-fold) is still able to reconstruct 62/83 Mb of the data set with high continuity and low error (0.24 errors/100 kb; 7.54 mismatches/100 kb; 129.77 indels/100 kb). Here, the coverage is insufficient for accurate polishing [97], leading to the highest indel error rate, the predominant error mode of the PacBio sequencer. This result could likely be improved by Illumina-based correction [85]. The remaining assemblies did not recover >28/83 Mb of the data set. However, the TSLR assembly had a low error rate across all metrics (0.19 errors/100 kb; 4.95 mismatches/100 kb; 1.65 indels/100 kb).

Conclusion

Here, we have highlighted recent developments in metagenomic assembly, both from a computational and technological perspective. Future technological advances are likely to have a significant impact on metagenomics. Our analysis on the mock HMP community highlights the power of single-molecule sequencing to resolve complex repeats and simplify assembly, albeit on a simple data set. Today, the use of long-read technologies in metagenomics applications is limited, as is the use of technologies generating long-range linking information. Some of these technologies, coupled with advances in

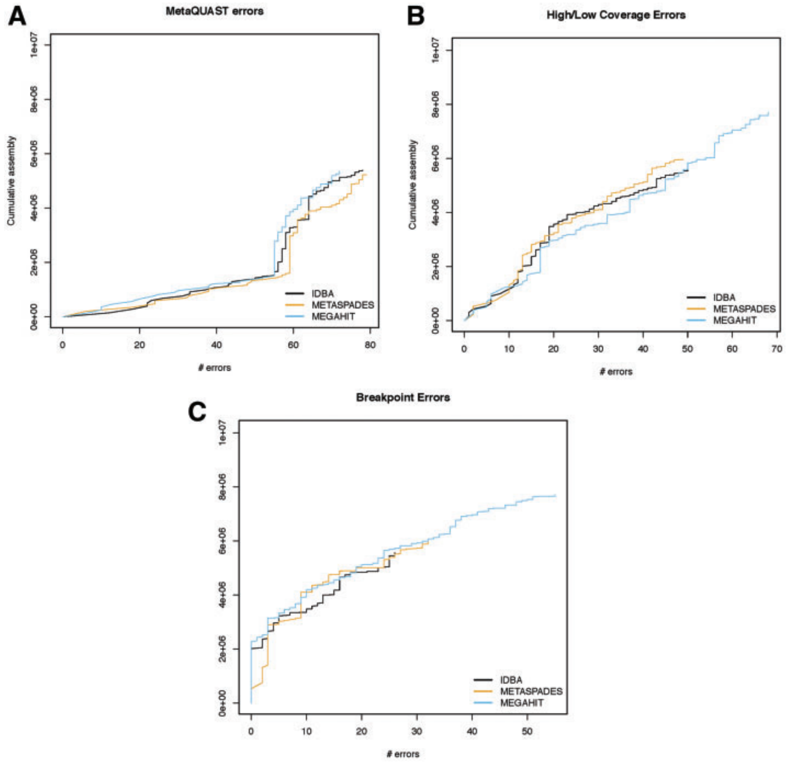


Figure 4. Reference-based and consistency-based evaluations of an HMP sample. FRC plots produced by MetaQUAST and VALET comparing assemblies of a stool sample (SRS016203) from the HMP (using IDBA-UD [24], metaSPAdes [9] and MEGAHIT [10]). (**A**) Reference-based (MetaQUAST) validation (all mis-assemblies flagged), (**B**) consistency-based (VALET) coverage anomalies and (**C**) consistency-based (VALET) break point errors. The *y*-axis represents the cumulative assembly size, considering contigs from largest to smallest. The *x*-axis represents the cumulative number of errors within the contigs comprising the corresponding *y*-axis value. Curves toward the top and left of the plot represent better assemblies—fewer errors for the same cumulative assembly size. Depending on metric, different assemblers perform best—MEGAHIT has highest consistency with reference genomes and fewest break points, while MetaSpades has fewer coverage anomalies.

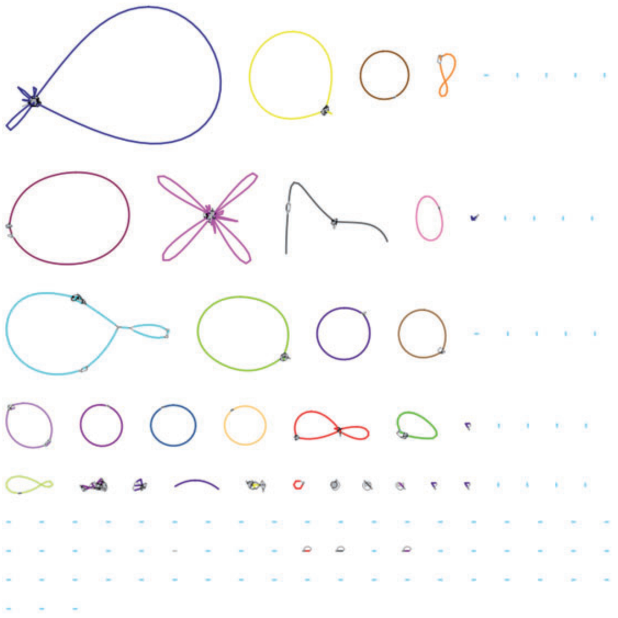


Figure 5. A bandage [95] visualization of the canu graphical assembly output. The unitigs in the assembly were aligned to the reference and assigned to their best match. Unitigs were colored by hand to match their species assignment. There are several large circular structures, which correspond to complete chromosomes. The smaller circles correspond to complete plasmids.

sample processing, will likely become sufficiently accurate and cost-effective, allowing for a much more accurate reconstruction of the genomic composition of microbial communities. Combinations of short-read (for coverage) and long-read (for continuity) are also possible, and several hybrid assemblers have been developed [98–102]. New technologies will also provide new capabilities, such as the ability of the Oxford Nanopore technology to ‘filter’ the reads within the sequencing instrument [103]. While only demonstrated with short targets, it has the potential to transform the way in which sequencing is used on diagnostic/detection applications, allowing researchers to bias the random sequencing process toward the DNA fragments of interest. New algorithms and software tools will continue to be developed to leverage such technological advances.

Effective assembly validation approaches are a critical need for the further progress of the field. Such tools will help researchers evaluate new software tools and sequencing strategies, and will highlight opportunities for further algorithmic development. Both reference-based and *de novo* validation strategies are important and need to continue to be developed. Reference-based methods provide a valuable lower bound on the performance of tools, while *de novo* methods allow the validation of assembly results and tuning of parameters even in settings where reference genomes are unavailable.

Key Points

- Despite recent advances in metagenomic assembly, assembled contigs are imperfect, and validation is key for moving forward.
- There are numerous metagenomic assembly metrics; understanding how to interpret each individual metric is key to evaluation of the accuracy of assembly of gene and genomes from metagenomes.
- VALET represents a *de novo* pipeline for detecting misassemblies in metagenomic data sets.
- Long-read technologies, such as PacBio RSII/Sequel, Oxford Nanopore, etc., are highly effective in the context of isolate genomes, but technical hurdles remain before they can be used routinely in metagenomic applications.

Acknowledgements

Opinions expressed in this article are of the authors' and do not necessarily reflect the policies and views of NIST or affiliated venues. Official contribution of NIST; not subject to copyrights in United States.

Funding

The NIH (grant numbers R01-HG-004885 and R01-AI-100947 to M.P.), and the NSF (grant numbers IIS-1117247 and IIS-0812111 to M.P.). The Intramural Research Program of the National Human Genome Research Institute, National Institutes of Health (to S.K.).

References

1. Podell S, Ugalde JA, Narasingarao P, *et al*. Assembly-driven community genomics of a hypersaline microbial ecosystem. *PLoS One* 2013;**8**:e61692.
2. Narasingarao P, Podell S, Ugalde JA, *et al*. *De novo* metagenomic assembly reveals abundant novel major lineage of Archaea in hypersaline microbial communities. *ISME J* 2012;**6**:81–93.
3. Ji P, Zhang Y, Wang J, *et al*. MetaSort untangles metagenome assembly by reducing microbial community complexity. *Nat Commun* 2017;**8**:14306.
4. Sangwan N, Xia F, Gilbert JA. Recovering complete and draft population genomes from metagenome datasets. *Microbiome* 2016;**4**:8.
5. Kingsford C, Schatz MC, Pop M. Assembly complexity of prokaryotic genomes using short reads. *BMC Bioinformatics* 2010; **11**:21.
6. Nagarajan N, Pop M. Parametric complexity of sequence assembly: theory and applications to next generation sequencing. *J Comput Biol* 2009;**16**:897–908.
7. Treangen TJ, Abraham AL, Touchon M, *et al*. Genesis, effects and fates of repeats in prokaryotic genomes. *FEMS Microbiol Rev* 2009;**33**:539–71.
8. Koren S, Harhay GP, Smith TP, *et al*. Reducing assembly complexity of microbial genomes with single-molecule sequencing. *Genome Biol* 2013;**14**:R101.
9. Nurk S, Meleshko D, Korobeynikov A, *et al*. metaSPAdes: a new versatile metagenomic assembler. *Genome Res* 2017;**27**:824–34.
10. Li D, Liu CM, Luo R, *et al*. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics* 2015;**31**: 1674–6.
11. Mikheenko A, Saveliev V, Gurevich A. MetaQUAST: evaluation of metagenome assemblies. *Bioinformatics* 2016;**32**: 1088–90.
12. Parks DH, Imelfort M, Skennerton CT, *et al*. CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. *Genome Res* 2015;**25**: 1043–55.
13. Pevzner PA, Tang H, Waterman MS. An Eulerian path approach to DNA fragment assembly. *Proc Natl Acad Sci USA* 2001;**98**:9748–53.
14. Namiki T, Hachiya T, Tanaka H, *et al*. MetaVelvet: an extension of velvet assembler to *de novo* metagenome assembly from short sequence reads. *Nucleic Acids Res* 2012;**40**:e155.
15. Sohn J, Nam JW. The present and future of *de novo* whole-genome assembly. *Brief Bioinform* 2016. doi: 10.1093/bib/bbw096.
16. Tomescu AI, Medvedev P. Safe and complete contig assembly through omnitigs. *J Comput Biol* 2017;**24**:590–602.
17. Nagarajan N, Pop M. Sequence assembly demystified. *Nat Rev Genet* 2013;**14**:157–67.
18. Miller JR, Koren S, Sutton G. Assembly algorithms for next-generation sequencing data. *Genomics* 2010;**95**:315–27.
19. Medvedev P, Scott E, Kakaradov B, *et al*. Error correction of high-throughput sequencing datasets with non-uniform coverage. *Bioinformatics* 2011;**27**:i137–41.
20. Song L, Florea L, Langmead B. Lighter: fast and memory-efficient sequencing error correction without counting. *Genome Biol* 2014;**15**:509.
21. Morowitz MJ, Denef VJ, Costello EK, *et al*. Strain-resolved community genomic analysis of gut microbial colonization in a premature infant. *Proc Natl Acad Sci USA* 2011;**108**:1128–33.
22. Salmela L, Walve R, Rivals E, *et al*. Accurate self-correction of errors in long reads using de Bruijn graphs. *Bioinformatics* 2017;**33**:799–806.
23. Greenwald WW, Klitgord N, Seguritan V, *et al*. Utilization of defined microbial communities enables effective evaluation of meta-genomic assemblies. *BMC Genomics* 2017;**18**:296.
24. Peng Y, Leung HCM, Yiu SM, *et al*. IDBA-UD: a *de novo* assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics* 2012;**28**:1420–8.
25. Peng Y, Leung HCM, Yiu SM, *et al*. IDBA—a practical iterative de Bruijn graph *de novo* assembler. In: *Proceedings of the 14th Annual International Conference on Research in Computational Molecular Biology (RECOMB 2010)*. 2010, 426–40. ACM Press.
26. Bowe A, Onodera T, Sadakane K, *et al*. Succinct de Bruijn graphs. In: *Proceedings of the 12th international conference on Algorithms in Bioinformatics (WABI 2012)*. 2012, 225–35. Springer.
27. Bankevich A, Nurk S, Antipov D, *et al*. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *J Comput Biol* 2012;**19**:455–77.
28. Prjibelski AD, Vasilinetc I, Bankevich A, *et al*. ExSPAnder: a universal repeat resolver for DNA fragment assembly. *Bioinformatics* 2014;**30**:i293–301.
29. Cai W, Aburatani H, Stanton VP, *et al*. Ordered restriction endonuclease maps of yeast artificial chromosomes created by optical mapping on surfaces. *Proc Natl Acad Sci USA* 1995; **92**:5164–8.

30. Gao S, Sung WK, Nagarajan N. Opera: reconstructing optimal genomic scaffolds with high-throughput paired-end sequences. *J Comput Biol* 2011;**18**:1681–91.
31. Gao S, Bertrand D, Chia BK, *et al*. OPERA-LG: efficient and exact scaffolding of large, repeat-rich eukaryotic genomes with performance guarantees. *Genome Biol* 2016;**17**:102.
32. Salmela L, Mäkinen V, Välimäki N, *et al*. Fast scaffolding with small independent mixed integer programs. *Bioinformatics* 2011;**27**:3259–65.
33. Pop M, Kosack DS, Salzberg SL. Hierarchical scaffolding with Bambus. *Genome Res* 2004;**14**:149–59.
34. Gnerre S, Maccallum I, Przybylski D, *et al*. High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proc Natl Acad Sci USA* 2011;**108**:1513–18.
35. Mouse Genome Sequencing Consortium; Waterston RH, Lindblad-Toh K, *et al*. Initial sequencing and comparative analysis of the mouse genome. *Nature* 2002;**420**:520–62.
36. Zheng GX, Lau BT, Schnall-Levin M, *et al*. Haplotyping germline and cancer genomes with high-throughput linked-read sequencing. *Nat Biotechnol* 2016;**34**:303–11.
37. McCoy RC, Taylor RW, Blauwkamp TA, *et al*. Illumina TruSeq synthetic long-reads empower *de novo* assembly and resolve complex, highly-repetitive transposable elements. *PLoS One* 2014;**9**:e106689.
38. Burton JN, Adey A, Patwardhan RP, *et al*. Chromosome-scale scaffolding of *de novo* genome assemblies based on chromatin interactions. *Nat Biotechnol* 2013;**31**:1119–25.
39. Kaplan N, Dekker J. High-throughput genome scaffolding from *in vivo* DNA interaction frequency. *Nat Biotechnol* 2013; **31**:1143–7.
40. Lieberman-Aiden E, van Berkum NL, Williams L, *et al*. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* 2009;**326**: 289–93.
41. Olm MR, Brown CT, Brooks B, *et al*. Identical bacterial populations colonize premature infant gut, skin, and oral microbiomes and exhibit different in situ growth rates. *Genome Res* 2017;**27**:601–12.
42. Baker BJ, Banfield JF. Microbial communities in acid mine drainage. *FEMS Microbiol Ecol* 2003;**44**:139–52.
43. Sharon I, Morowitz MJ, Thomas BC, *et al*. Time series community genomics analysis reveals rapid shifts in bacterial species, strains, and phage during infant gut colonization. *Genome Res* 2013;**23**:111–20.
44. Mackelprang R, Waldrop MP, DeAngelis KM, *et al*. Metagenomic analysis of a permafrost microbial community reveals a rapid response to thaw. *Nature* 2011;**480**:368–71.
45. Schloissnig S, Arumugam M, Sunagawa S, *et al*. Genomic variation landscape of the human gut microbiome. *Nature* 2013;**493**:45–50.
46. Eppley JM, Tyson GW, Getz WM, *et al*. Strainer: software for analysis of population variation in community genomic datasets. *BMC Bioinformatics* 2007;**8**:398.
47. Koren S, Treangen TJ, Pop M. Bambus 2: scaffolding metagenomes. *Bioinformatics* 2011;**27**:2964–71.
48. Nijkamp JF, Pop M, Reinders MJ, *et al*. Exploring variation-aware contig graphs for (comparative) metagenomics using MaryGold. *Bioinformatics* 2013;**29**:2826–34.
49. Gutwenger C, Mutzel P. A linear time implementation of SPQR-trees. In: *Proceedings of the 8th International Symposium on Graph Drawing* (LNCS, volume 1984). 2001, 70–90. Springer.
50. Eren AM, Esen ÖC, Quince C, *et al*. Anvi'o: an advanced analysis and visualization platform for 'omics data. *PeerJ* 2015;**3**: e1319.
51. Grabherr MG, Haas BJ, Yassour M, *et al*. Trinity: reconstructing a full-length transcriptome without a genome from RNA-Seq data. *Nat Biotechnol* 2011;**29**:644–52.
52. Schneider GF, Dekker C. DNA sequencing with Nanopores. *Nat Biotechnol* 2012;**30**:326–8.
53. Eid J, Fehr A, Gray J, *et al*. Real-time DNA sequencing from single polymerase molecules. *Science* 2009;**323**:133–8.
54. http://www.pacb.com/products-and-services/pacbio-systems.
55. https://nanoporetech.com/products.
56. Voskoboynik A, Neff NF, Sahoo D, *et al*. The genome sequence of the colonial chordate, *Botryllus schlosseri*. *Elife* 2013;**2**:e00569.
57. Bankevich A, Pevzner PA. TruSPAdes: barcode assembly of TruSeq synthetic long reads. *Nat Methods* 2016;**13**: 248–50.
58. Koren S, Phillippy AM. One chromosome, one contig: complete microbial genomes from long-read sequencing and assembly. *Curr Opin Microbiol* 2015;**23**:110–20.
59. Kuleshov V, Jiang C, Zhou W, *et al*. Synthetic long-read sequencing reveals intraspecies diversity in the human microbiome. *Nat Biotechnol* 2016;**34**:64–9.
60. Bickhart DM, Rosen BD, Koren S, *et al*. Single-molecule sequencing and chromatin conformation capture enable *de novo* reference assembly of the domestic goat genome. *Nat Genet* 2017;**49**:643–50.
61. Pendleton M, Sebra R, Pang AW, *et al*. Assembly and diploid architecture of an individual human genome via single-molecule technologies. *Nat Methods* 2015;**12**:780–6.
62. Berlin K, Koren S, Chin CS, *et al*. Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nat Biotechnol* 2015;**33**:623–30.
63. Gordon D, Huddleston J, Chaisson MJ, *et al*. Long-read sequence assembly of the gorilla genome. *Science* 2016;**352**:aae0344.
64. Zimin AV, Puiu D, Luo MC, *et al*. Hybrid assembly of the large and highly repetitive genome of Aegilops tauschii, a progenitor of bread wheat, with the MaSuRCA mega-reads algorithm. *Genome Res* 2017;**27**:787–92.
65. Seo JS, Rhie A, Kim J, *et al*. *De novo* assembly and phasing of a Korean human genome. *Nature* 2016;**538**:243–7.
66. Jarvis DE, Ho YS, Lightfoot DJ, *et al*. The genome of *Chenopodium quinoa*. *Nature* 2017;**542**:307–12.
67. Jain M, Koren S, Quick J, *et al*. Nanopore sequencing and assembly of a human genome with ultra-long reads. *bioRxiv* 2017;128835.
68. Koren S, Walenz BP, Berlin K, *et al*. Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome Res* 2017;**27**:722–36.
69. Chin CS, Peluso P, Sedlazeck FJ, *et al*. Phased diploid genome assembly with single-molecule real-time sequencing. *Nat Methods* 2016;**13**:1050–4.
70. Li H. Minimap and miniasm: fast mapping and *de novo* assembly for noisy long sequences. *Bioinformatics* 2016;**32**: 2103–10.
71. Dudchenko O, Batra SS, Omer AD, *et al*. *De novo* assembly of the *Aedes aegypti* genome using Hi-C yields chromosome-length scaffolds. *Science* 2017;**356**:92–5.
72. White RA, Bottos EM, Roy Chowdhury T, *et al*. Moleculo long-read sequencing facilitates assembly and genomic binning from complex soil metagenomes. *mSystems* 2016;**1**: e00045–16.
73. Kuleshov V, Snyder MP, Batzoglou S. Genome assembly from synthetic long read clouds. *Bioinformatics* 2016;**32**: i216–24.

74. Driscoll CB, Otten TG, Brown NM, *et al*. Towards long-read metagenomics: complete assembly of three novel genomes from bacteria dependent on a diazotrophic cyanobacterium in a freshwater lake co-culture. *Stand Genomic Sci* 2017;**12**:9.
75. Tsai YC, Conlan S, Deming C, *et al*. Resolving the complexity of human skin metagenomes using single-molecule sequencing. *MBio* 2016;**7**:e01948–15.
76. Olson ND, Morrow JB. DNA extract characterization process for microbial detection methods development and validation. *BMC Res Notes* 2012;**5**:668.
77. Nair S, Karim R, Cardosa MJ, *et al*. Convenient and versatile DNA extraction using agarose plugs for ribotyping of problematic bacterial species. *J Microbiol Methods* 1999;**38**:63–7.
78. Maydan J, Thomas M, Tabanfar L, *et al*. Electrophoretic high molecular weight DNA purification enables optical mapping. *J Biomol Tech* 2013;**24**:S57.
79. Tighe S, Afshinnekoo E, Rock TM, *et al*. Genomic methods and microbiological technologies for profiling novel and extreme environments for the Extreme Microbiome Project (XMP). *J Biomol Tech* 2017;**28**:31–9.
80. Staden R. A strategy of DNA sequencing employing computer programs. *Nucleic Acids Res* 1979;**6**:2601–10.
81. Rahman A, Pachter L. CGAL: computing genome assembly likelihoods. *Genome Biol* 2013;**14**:R8.
82. Ghodsi M, Hill CM, Astrovskaya I, *et al*. *De novo* likelihood-based measures for comparing genome assemblies. *BMC Res Notes* 2013;**6**:334.
83. Hill CM, Astrovskaya I, Huang H, *et al*. *De novo* likelihood-based measures for comparing metagenomic assemblies. In: *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine, 2013*. 2013, 94–8.
84. Clark SC, Egan R, Frazier PI, *et al*. ALE: a generic assembly likelihood evaluation framework for assessing the accuracy of genome and metagenome assemblies. *Bioinformatics* 2013;**29**:435–43.
86. Gurevich A, Saveliev V, Vyahhi N, *et al*. QUAST: quality assessment tool for genome assemblies. *Bioinformatics* 2013;**29**:1072–5.
87. Lander ES, Waterman MS. Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics* 1988;**2**:231–9.
88. Phillippy AM, Schatz MC, Pop M. Genome assembly forensics: finding the elusive mis-assembly. *Genome Biol* 2008;**9**:R55.
89. Narzisi G, Mishra B. Comparing *de novo* genome assembly: the long and short of it. *PLoS One* 2011;**6**:e19175.
90. Vezzi F, Narzisi G, Mishra B. Reevaluating assembly evaluations with feature response curves: GAGE and assemblathons. *PLoS One* 2012;**7**:e52210.
91. Hunt M, Kikuchi T, Sanders M, *et al*. REAPR: a universal tool for genome assembly evaluation. *Genome Biol* 2013;**14**:R47.
85. Walker BJ, Abeel T, Shea T, *et al*. Pilon: an integrated tool for comprehensive microbial variant detection and genome assembly improvement. *PLoS One* 2014;**9**:e112963.
92. Hill CM. Novel methods for comparing and evaluating single and metagenomic assemblies. PhD Thesis, University Maryland, 2015. http://drum.lib.umd.edu/handle/1903/17100
93. Thorvaldsdóttir H, Robinson JT, Mesirov JP. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. *Brief Bioinform* 2013;**14**:178–92.
94. DevNet. Human Microbiome Project MockB Shotgun. 2014. https://github.com/PacificBiosciences/DevNet/wiki/Human_Microbiome_Project_MockB_Shotgun (28 July 2017, date last accessed).
95. Wick RR, Schultz MB, Zobel J, *et al*. Bandage: interactive visualization of *de novo* genome assemblies. *Bioinformatics* 2015;**31**:3350–2.
96. Chin CS, Alexander DH, Marks P, *et al*. Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. *Nat Methods* 2013;**10**:563–9.
97. Alexander DH. Quiver FAQ. 2016. https://github.com/PacificBiosciences/GenomicConsensus/blob/master/doc/FAQ.rst
98. Madoui MA, Engelen S, Cruaud C, *et al*. Genome assembly using Nanopore-guided long and error-free DNA reads. *BMC Genomics* 2015;**16**:327.
99. Koren S, Schatz MC, Walenz BP, *et al*. Hybrid error correction and *de novo* assembly of single-molecule sequencing reads. *Nat Biotechnol* 2012;**30**:693–700.
100. Antipov D, Korobeynikov A, McLean JS, *et al*. hybridSPAdes: an algorithm for hybrid assembly of short and long reads. *Bioinformatics* 2016;**32**:1009–15.
101. Ye C, Hill CM, Wu S, *et al*. DBG2OLC: efficient assembly of large genomes using long erroneous reads of the third generation sequencing technologies. *Sci Rep* 2016;**6**:31900.
102. Goodwin S, Gurtowski J, Ethe-Sayers S, *et al*. Oxford Nanopore sequencing, hybrid error correction, and *de novo* assembly of a eukaryotic genome. *Genome Res* 2015;**25**:1750–6.
103. Loose M, Malla S, Stout M. Real-time selective sequencing using Nanopore technology. *Nat Methods* 2016;**13**:751–4.