

1 Evaluating Metagenome Assembly on a Simple
2 Defined Community with Many Strain Variants

3 Sherine Awad¹, Luiz Irber¹, C. Titus Brown^{1*}
1 Department of Population Health and Reproduction
University of California, Davis
Davis, CA 95616 USA
* E-mail: ctbrown@ucdavis.edu

4 June 24, 2017

5 **Abstract**

6 We evaluate the performance of three metagenome assemblers, IDBA,
7 SPAdes, and MEGAHIT, on short-read sequencing of a defined “mock”
8 community from Shakya et al. (2013) containing 64 genomes. We up-
9 date the reference metagenome for this mock community and detect
10 several additional genomes in the read data set. We show that strain
11 confusion results in significant loss of reference genomes that are oth-
12 erwise completely present in the read data set. In agreement with
13 previous studies, we find that MEGAHIT performs best computation-
14 ally; we also show that MEGAHIT tends to recover larger portions of
15 the strain variants than the other assemblers.

16 Introduction

17 Metagenomics refers to sequencing of DNA from a mixture of organisms,
18 often from an environmental or uncultured sample. Unlike whole genome
19 sequencing, metagenomics targets a mixture of genomes, which introduces
20 metagenome-specific challenges in analysis [1]. Most approaches to analyz-
21 ing metagenomic data rely on mapping or comparing sequencing reads to
22 reference sequence collections. However, reference databases contain only
23 a small subset of microbial diversity [2], and the much of the remaining
24 diversity is evolutionarily distant and search techniques may not access it.

25 As sequencing capacity increases and sequence data is generated from
26 many more environmental samples, metagenomics is increasingly using de
27 novo assembly techniques to generate new reference genomes and metagenomes.
28 There are a number of metagenome assemblers that are widely used. How-
29 ever, evaluating the results of these assemblers is challenging due to the
30 general lack of good quality reference metagenomes.

31 Moya et al. in [3] evaluated metagenome assembly using two simulated
32 454 viral metagenome and six assemblers. The assemblies were evaluated
33 based on several metrics including N50, percentages of reads assembled, ac-
34 curacy when compared to the reference genome. In addition to, chimeras per
35 contigs and the effect of assembly on taxonomic and functional annotations.

36 Mavromatis et al. in [4] provided a benchmark study to evaluate the
37 fidelity of metagenome process methods. The study used simulated metage-
38 nomic data sets constructed at different complexity levels. The datasets
39 were assembled using Phrap v3.57, Arachne v.2 [5] and JAZZ [6]. This
40 study evaluates assembly, gene prediction, and binning methods. However,
41 the study did not evaluate the assembly quality against a reference genome.

42 Rangwala et al. in [7] presented an evaluation study of metagenome
43 assembly. The study used a de Bruijn graph based assembler ABYSS [8] to
44 assemble simulated metagnome reads of 36 bp. The data set is classified at
45 different complexity levels. The study compares the quality of the assembly
46 of the data sets in terms of quality measures of contigs length, assembly
47 accuracy. The study also took into consideration the effect of kmer size and
48 the degree of chimericity. However, the study evaluated the assembly based
49 on one assembler, and did not evaluate assembly against several assemblers.
50 Also, both previous studies used simulated data, which may lack confounders
51 of assembly such as sequencing artifacts and GC bias.

52 Shakya et al. (2013) constructed a synthetic community of organisms by
53 mixing DNA isolated from individual cultures of 64 bacteria and archaea,

54 including a variety of strains across a range of nucleotide distances [9]. In
55 addition to performing 16s amplicon analysis and doing 454 sequencing, the
56 authors shotgun-sequenced the mixture with Illumina. While the authors
57 concluded that this metagenomic sequencing generally outperformed ampli-
58 con sequencing, they did not conduct an assembly based analysis. This data
59 set was also used in several other evaluation studies, including gbtools for
60 binning [10] and benchmarking of the MEGAHIT assembler [11].

61 More recently, several benchmark studies systematically evaluated metagenome
62 assembly of short reads. The Critical Assessment of Metagenome Interpre-
63 tation (CAMI) collaboration benchmarked a number of metagenome assem-
64 blers on several data sets of varying complexity, evaluating recovery of novel
65 genomes and multiple strain variants [12]. Notably, CAMI concluded that
66 “The resolution of strain-level diversity represents a substantial challenge to
67 all evaluated programs.” Another recent study evaluated eight assemblers
68 on nine environmental metagenomes and three simulated data sets [13]. Also
69 see [14].

70 In this study, we extend previous work by delving into questions of
71 chimeric misassembly and strain recovery in the Shakya et al. (2013) data
72 set. First, we update the list of reference genomes for Shakya et al. to in-
73 clude the latest Genbank assemblies. We then compare IDBA [15], SPAdes
74 [16], and MEGAHIT [17] performance on assembling this short-read data
75 set, and explore concordance in recovery between the three assemblers. We
76 describe the effects of “strain confusion” between multiple strains. We also
77 detect and analyze several previously unreported strains and genomes in
78 the Shakya et al. data set. We find that in the absence of closely related
79 genomes, all three metagenome assemblers recover 95% or more of known
80 reference genomes. However, in the presence of closely related genomes,
81 metagenome assemblers vary widely in their performance and can fail to re-
82 cover the majority of some genomes even when they are completely present
83 in the reads. Our report provides strong guidance on choice of assemblers
84 and extends previous analyses of this low-complexity metagenome bench-
85 marking data set.

86 Datasets

87 We used a diverse mock community data set constructed by pooling DNA
88 from 64 species of bacteria and archaea and sequencing them with Illumina
89 HiSeq. The raw data set consisted of 109,629,496 reads from Illumina HiSeq
90 101 bp paired-end sequencing (2x101) with an untrimmed total length of

91 11.07 Gbp and an estimated fragment size of 380 bp [9].

92 The original reads are available through the NCBI Sequence Read Archive
93 at Accession SRX200676. We updated the 64 reference genomes sets from
94 NCBI Genbank using the latest available assemblies (June 2017); updated
95 data is available for download at <https://osf.io/8uxj9/>.

96 **Methods**

97 The analysis code and run scripts for this paper are available at: [https://github.com/dib-](https://github.com/dib-lab/2015-metagenome-assembly/)
98 [lab/2015-metagenome-assembly/](https://github.com/dib-lab/2015-metagenome-assembly/). The scripts and overall pipeline were ex-
99 amined by the first and senior authors for correctness. In addition, the
100 bespoke reference-based analysis scripts were tested by running them on a
101 single-colony *E. coli* MG1655 data set with a high quality reference genome
102 [18].

103 **Quality Filtering**

104 We removed adapters with Trimmomatic v0.30 in paired-end mode with the
105 Truseq adapters [19], using light quality score trimming as recommended in
106 MacManes, 2014 [20].

107 **Reference Coverage Profile**

108 To evaluate how much of the reference metagenome was contained in the
109 read data, we used `bwa aln` (v0.7.7.r441) to map reads to the reference
110 genome [21]. We then calculated how many reference bases were covered by
111 mapped reads (custom script `coverage-profile.py`).

112 **Measuring k-mer inclusion and Jaccard similarity**

113 We used MinHashing as implemented in sourmash to estimate k-mer inclu-
114 sion and Jaccard similarity between data sets [22]. MinHash signatures were
115 prepared with ‘sourmash compute’ using ‘-scaled 10000’. K-mer inclusion
116 was computed by taking the ratio of the number of intersecting hashes with
117 the query over the total number of hashes in the subject MinHash. Jac-
118 card similarity was computed as in [23] by taking the ratio of the number
119 of intersecting hashes between the query and subject over the number of
120 hashes in the union. K-mer sizes for comparison were chosen at 21, 31, or

121 51, depending on the level of taxonomic specificity desired - genus, species,
122 or strain, as described in [24].

123 When specified, high-abundance k-mers were selected for counting by
124 using the script `trim-low-abund.py` script with `-C 5` from khmer 2.x [25,
125 26].

126 Assemblers

127 We assembled the quality-filtered reads using three different assemblers:
128 IDBA-UD [15], MetaSPAdes [16], and MEGAHIT [17]. For IDBA-UD v1.1.1
129 [15], we used `--pre-correction` to perform pre-correction before assembly
130 and `-r` for the pe files.

131 For MetaSPAdes v3.9.0 [16], we used `--meta --pe1-12 --pe1-s` where
132 `--meta` is used for metagenomic data sets, `--pe1-12` specifies the interlaced
133 reads for the first paired-end library, and `--pe1-s` provides the orphan reads
134 remaining from quality trimming.

135 For MEGAHIT v1.1.1-2-g02102e1 [17], we used `-l 101 -m 3e9 --cpu-only`
136 where `-l` is for maximum read length, `-m` is for max memory in bytes to
137 be used in constructing the graph, and `--cpu-only` to use only the CPU
138 and no GPUs. We also used `--presets meta-large` for large and complex
139 metagenomes, and `--12` and `-r` to specify the interleaved-paired-end and
140 single-end files respectively. MEGAHIT allows the specification of a memory
141 limit and we used `-M 1e+10` for 10 GB.

142 All three assemblies were executed on the same high-memory buy-in
143 node on the Michigan State University High Performance Compute Cluster,
144 and we recorded RAM and CPU time of each assembly job using the `qstat`
145 utility at the end of each run.

146 Unless otherwise mentioned, we eliminated all contigs less than 500 bp
147 from each assembly prior to further analysis.

148 Mapping

149 We aligned all quality-filtered reads to the reference metagenome with `bwa`
150 `aln` (v0.7.7.r441) [21]. We aligned paired-end and orphaned reads separately.
151 We then used `samtools` (v0.1.19) [27] to convert SAM files to BAM files for
152 both paired-end and orphaned reads. To count the unaligned reads, we
153 included only those records with the “4” flag in the SAM files [27].

154 **Assembly analysis using Nucmer**

155 We used the NUCmer tool from MUMmer3.23 [28] to align assemblies to the
156 reference genome with options `-coords -p`. Then we parsed the generated
157 “coords” file using a custom script `analyze_assembly.py`, and calculated
158 several analysis metrics across all three assemblies at a 99% alignment iden-
159 tity.

160 **Reference-based analysis of the assemblies**

161 We conducted reference-based analysis of the assemblies under two condi-
162 tions. “Loose” alignment conditions used all available alignments, including
163 redundant and overlapping alignments. “Strict” alignment conditions took
164 only the longest alignment for any given contig, eliminating all other align-
165 ments.

166 The script `summarize-coords2.py` was used to calculate aligned cov-
167 erage from the loose alignment conditions: each base in the reference was
168 marked as “covered” if it was included in at least one alignment. The script
169 `analyze_ng50.py` was used to calculate NGA 50 for each individual refer-
170 ence genome.

171 **Analysis of chimeric misassemblies**

172 We analyzed each assembly for chimeric misassemblies by counting the num-
173 ber of contigs that contained matches to two distinct reference genomes. In
174 order to remove secondary alignments from consideration, we included only
175 the longest non-overlapping NUCmer alignments for each contig at a mini-
176 mum alignment identity of 99%. We then used the script `analyze_chimeric2.py`
177 to find individual contigs that matched more than one distinct reference
178 genome. As a negative control on our analysis, we verified that this ap-
179 proach yielded no positive results when applied to the alignments of the
180 reference metagenome against itself.

181 **Results**

182 **The raw data is high quality.**

183 The reads contains 11,072,579,096 bp (11.07 Gbp) in 109,629,496 reads with
184 101.0 average length (2x101bp Illumina HiSeq).

Table 1: Jaccard containment of the reference in the reads

k-mer size	% reference in reads
21	96.8%
31	95.9%
41	94.9%
51	94.1%

Trimming removed 686,735 reads (0.63%). After trimming, we retained 108,422,358 paired reads containing 10.94 Gbp with an average length of 100.9 bases. A total of 46.56 Mbp remained in 520,403 orphan reads with an average length of 89.5 bases. In total, the quality trimmed data contained 10.98 Gbp in 108,942,761 reads. This quality trimmed (“QC”) data set was used as the basis for all further analyses.

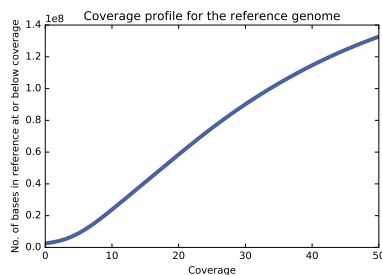


Figure 1: Cumulative coverage profile for the reference metagenome, based on read mapping.

The reference metagenome is not completely present in the reads.

We next evaluated the fraction of the reference genome covered by at least one read (see Methods for details). Quality filtered reads cover 203,058,414 (98.76%) bases of the reference metagenome (205,603,715 bp total size). Figure 1 shows the cumulative coverage profile of the reference metagenome, and the percentage of bases with that coverage. Most of the reference metagenome was covered at least minimally; only 3.33% of the reference metagenome had mapping coverage <5 , and 1.24% of the bases in the reference were not covered by any reads in the QC data set.

In order to evaluate reconstructability with De Bruijn graph assemblers,

we next examined k-mer containment of the reference in the reads for k of 21, 31, 41, and 51 (Table 1). The k-mer overlap decreases from 96.8% to 94.1% as the k-mer size increases. This could be caused by low coverage of some portions of the reference and/or variation between the reads and the reference.

Some individual reference genomes are poorly represented in the reads.

Table 2: Top uncovered genomes

Genome	Read coverage	21-mer presence
<i>B. bronchiseptica</i>	98.2%	97.3%
<i>D. vulgaris</i> DP4	93.2%	82.5%
<i>T. thermophilus</i> HB27	91.1%	79.7%
<i>E. faecalis</i> V583	74.6%	65.6%
<i>F. nucleatum</i>	47.6%	18.2%

To see if specific reference genomes exhibited low coverage, we analyzed read mapping coverage and 21-mer containment for individual genomes. Of the 64 reference genomes used in the metagenome, 59 had a per-base mapping coverage above 95% and a 21-mer containment in the QC reads above 95%. The remaining five varied significantly in both metrics (Table 4), with *F. nucleatum* the lowest – only 47.6% of the bases in the reference genome are covered by one or more mapped reads, and only 18.2% of the 21-mers in the *F. nucleatum* reference genome are present in the reads at any abundance.

We next did a 51-mer containment analysis of each reference genome in the reads, mimicking the analysis done in [24]. 99% or more of the constituent 51-mers for 51 of the 64 reference genomes were present in the reads, suggesting that each of the 51 genomes was entirely present at some minimal coverage.

We excluded the remaining 13 genomes (see Table 3) from any comparative analysis of assembly quality, because interpreting coverage and misassembly analysis for these genomes would be challenging.

Table 3: Genomes removed from reference for low 51-mer presence

51-mers in reads	Genome
98.7	<i>Leptothrix cholodnii</i>
98.7	<i>Haloferax volcanii</i> DS2
98.6	<i>Salinispora tropica</i> CNB-440
97.4	<i>Deinococcus radiodurans</i>
97.2	<i>Zymomonas mobilis</i>
97.1	<i>Ruegeria pomeroyi</i>
96.8	<i>Shewanella baltica</i> OS223
95.5	<i>B. bronchiseptica</i> D989
94.5	<i>Burkholderia xenovorans</i>
72.0	<i>Desulfovibrio vulgaris</i> DP4
65.0	<i>Thermus thermophilus</i> HB27
53.4	<i>Enterococcus faecalis</i>
4.7	<i>Fusobacterium nucleatum</i> ATCC 25586

Table 4: Running Time and Memory Utilization

Assembler	CPU time	Wall time	RAM
MEGAHIT	52hr 25m	4 hr 9m	11.4 GB
IDBA-UD	17h		149.1 GB
SPAdes	94hr 43m	94hr 44m	100.7 GB

MEGAHIT is the fastest and lowest-memory assembler evaluated

We ran three commonly used metagenome assemblers on the QC data set: IDBA-UD, SPAdes, and MEGAHIT. We recorded the time and memory usage of each (Table 4). In computational requirements, MEGAHIT outperformed both SPAdes and IDBA-UD considerably, producing an assembly in four hours – approximately 4 times faster than IDBA and 8 times faster than SPAdes. MEGAHIT used only 11.4 GB of RAM – 1/13th to 1/9th the memory used by IDBA and SPAdes, respectively.

The assemblies contain most of the raw data

We assessed read inclusion in assemblies by mapping the QC reads to the length-filtered assemblies and counting the remaining unmapped reads. Depending on the assembly, between 2.7 million and 3.9 million reads (2.5-

Table 5: Read and high-abundance (> 5) k-mer exclusion from assemblies

Assembly	Unmapped Reads	51-mers omitted
IDBA	3,328,674 (3.05%)	2.4%
SPAdes	3,844,123 (3.52%)	3.2%
MEGAHIT	2,737,640 (2.51%)	2.8%

3.5%) did not map to the assemblies (Table 5). All of the assemblies included the large majority of high-abundance 51-mers (more than 96.8% in all cases).

Much of the reference is covered by the assemblies.

Table 6: Contig coverage of reference with loose alignment conditions.

Assembly	bases aligned	duplication	51-mers
MEGAHIT	96.2%	0.72%	96.7%
SPAdes	95.8%	0.99%	96.2%
IDBA	95.6%	0.88%	97.2%

We next evaluated the extent to which the assembled contigs recovered the “known/true” metagenome sequence by aligning each assembly to the adjusted reference (Table 6). Each of the three assemblers generates contigs that cover more than 95.6% of the reference metagenome at high identity (99%) with little duplication (0.72-0.99%). All three assemblies contain between 96.2% and 97.2% of the 51-mers in the reference.

At 99% identity with the loose mapping approach, approximately 1.8% of the reference is missed by all three assemblers, while 0.9% is uniquely covered by MEGAHIT, 0.6% is uniquely covered by SPAdes, and 0.4% is uniquely covered by IDBA.

The generated contigs are broadly accurate.

When counting only the best (longest) alignment per contig at a 99% identity threshold, each of the three assemblies recovers more than 87.3% of the reference, with MEGAHIT recovering the most – 93.8% of the reference (Table 7).

Table 7: Contig accuracy measured by reference coverage with strict alignment.

Assembly	% covered
MEGAHIT	93.8%
IDBA	89.5%
SPAdes	87.3%

Individual genome statistics vary widely in the assemblies.

We computed the NGA50 for each individual genome and assembly in order to compare assembler performance on genome recovery (see left panel of Figure 2). The NGA50 statistics for individual genomes vary widely, but there are consistent assembler-specific trends: IDBA yields the lowest NGA50 for 28 of the 51 genomes, while SPAdes yields the highest NGA50 for 32 of the 51 genomes.

We also evaluated aligned coverage per genome for each of the three assemblies (right panel, Figure 2). We found that a 13 of the 51 genomes were missing 5% or more of bases in at least one assembly, despite all 51 genomes having 99% or higher read- and 51-mer coverage.

There are 12 genomes with k=31 Jaccard similarity greater than 2% to other genomes in the community, and these (denoted by '*' after the name) typically had lower NGA50 and aligned coverage numbers than other genomes. In particular, these constituted 12 of the 13 genomes missing 5% or more of their content, and the lowest eight NGA50 numbers.

Longer contigs are less likely to be chimeric.

Table 8: Chimeric contigs by contig length.

Assembly	> 50kb	> 5kb	> 500 bp
IDBA	0	1	7
MEGAHIT	1	4	14
SPAdes	0	3	30

Chimerism is the formation of contigs that include sequence from multiple genomes. We evaluated the rate of chimerism in contigs at three different contig length cutoffs: 500bp, 5kb, and 50kb (Table 8). We found that the percentage of contigs that match to the genomes of two or more different species drop as the minimum contig size increases, to the point where only

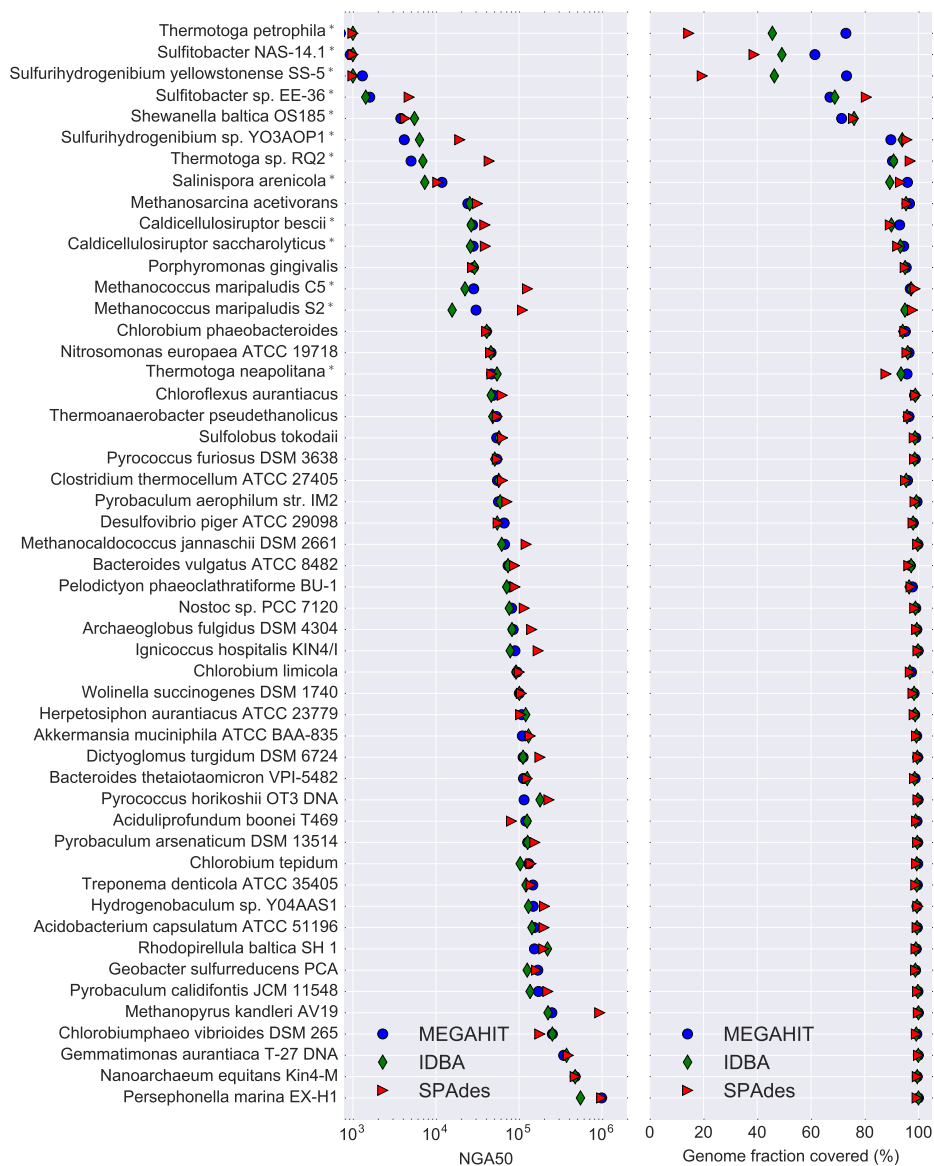


Figure 2: NGA50 by genome and assembler. A '*' after the name indicates the presence of at least one other genome with > 2% Jaccard similarity at k=31 in the community.

Table 9: Genbank genomes detected in assembly of unmapped reads

match	Genbank genome
44.1%	<i>Fusobacterium</i> sp. <i>OBRC1</i>
23.0%	<i>P. ruminis</i> strain <i>ML2</i>
18.2%	<i>Thermus thermophilus</i> <i>HB8</i>
7.7%	<i>P. ruminis</i> strain <i>CGMCC</i>
8.2%	<i>Enterococcus faecalis</i> <i>M7</i>
7.3%	<i>F. nucleatum</i> <i>13-3C</i>
3.7%	<i>F. nucleatum</i> subsp. <i>polymorphum</i>
2.9%	<i>Fusobacterium</i> <i>hwasookii</i>
1.0%	<i>E. coli</i> isolate <i>YS</i>
1.7%	<i>F. nucleatum</i> subsp. <i>polymorphum</i>
1.9%	<i>F. nucleatum</i> subsp. <i>vincentii</i>

the MEGAHIT assembly had a single chimeric contig longer than 50kb. Overall, chimeric misassemblies were rare, with no assembler generating more than 30 chimeric contigs out of thousands of total contigs.

The unmapped reads contain strain variants of reference genomes.

Approximately 4.8 million reads (4.4%) from the QC data set did not map anywhere in the reference provided by the authors of [9]. We extracted and assembled these reads in isolation using MEGAHIT, yielding 6.5 Mbp of assembly in 1711 contigs > 500bp in length. We then did a k-mer inclusion analysis of this assembly against all of the Genbank genomes at k=31, and estimated the fraction of the k-mers that belonged to different species (Table 9). We find that 51.1% of the k-mer content of these contigs positively match to a genome present in Genbank but not in the reference metagenome.

To verify these assignments, we aligned the MEGAHIT assembly of unmapped reads to the Genbank genomes in Table 9 with nucmer using “loose” alignment criteria. We found that 1.78 Mbp of the contigs aligned at 99% identity or better to these Genbank genomes. We also confirmed that, as expected, there are no matches in this assembly to the full updated reference metagenome.

We note that all but the two *P. ruminis* matches and the *E. coli* isolate YS are strain variants of species that are part of the defined community but are not completely present in the reads (see Table 2). For *Proteiniclasticum ruminis*, there is no closely related species in the mock community

302 design, and very little of the MEGAHIT assembly aligns to known *P. ru-*
303 *minis* genomes at 99%. However, there are many alignments to *P. ruminis*
304 at 94% or higher, for approximately 2.73 Mbp total. This suggests that the
305 unmapped reads contain at least some data from a novel species of *Proteini-*
306 *clasticum*.

307 Discussion

308 Assembly recovers basic content sensitively and accurately.

309 All three assemblers performed well in assembling contigs from the con-
310 tent that was fully present in reads and k-mers. After length filtering,
311 all three assemblies contained more than 95% of the reference (Table 6);
312 even with removal of secondary alignments, more than 87% was recovered
313 by each assembler (Table 7). About half the constituent genomes had an
314 NGA50 of 50kb or higher (Figure 2), which, while low for current Illumina
315 single-genome sequencing, is sufficient to recover operon-level relationships
316 for many genes.

317 The presence of multiple closely related genomes confounds 318 assembly.

319 As reported by CAMI, we also find that the presence of closely related
320 genomes in the metagenome causes many assembly problems. This is clearly
321 shown by Figure 2, where 12 of the bottom 14 genomes by NGA50 (left
322 panel) also exhibit poor genome recovery by assembly (right panel). Inter-
323 estingly, different assemblers handle this quite differently, with e.g. SPAdes
324 failing to recover essentially any of *Thermotoga petrophila*, while MEGAHIT
325 recovers 73%. The presence of nearby genomes is an almost perfect predic-
326 tor that one or more assembler will fail to recover 5% or more - of the 13/51
327 genomes for which less than 95% is recovered, 12 of them have close genomes
328 in the community. Interestingly, very little similarity is needed - all genomes
329 with Jaccard similarity of 2% or higher at k=31 exhibited these problems.

330 The *Shewanella baltica* OS185 genome is a good example: there are two
331 strain variants, OS185 and OS223, present in the defined community. Both
332 are present at more than 99% in the reads, and more than 98% in 51-mers,
333 but only 75% of *S. baltica* OS185 and 50% of *S. baltica* OS223 are recovered
334 by assemblers. This is a clear case of “strain confusion” where the assemblers
335 simply fail to output contigs for a substantial portion of the two genomes.

336 Another interest of this study was to examine cross-species chimeric
337 assembly, in which a single contig is formed from multiple genomes. In
338 Table 8, we show that there is relatively little cross-species chimerism.

339 **MEGAHIT performs best by several metrics.**

340 MEGAHIT is clearly the most efficient computationally, outperforming both
341 SPAdes and IDBA by 5-10x in memory and 17-42x in time (Table 4). The
342 MEGAHIT assembly also included more of the reads than either IDBA or
343 SPAdes, and omitted only 0.4% more of the unique 51-mers from the reads
344 than IDBA. MEGAHIT covered more of the reference genome with both
345 loose and strict alignments (Table 6 and Table 7), with little duplication.
346 This is clearly because of MEGAHIT’s superior performance in recovering
347 the genomes of closely related strains (Figure 2, right panel). The sum
348 “fraction of genome recovered” is arguably the most important measure of
349 a metagenome assembler (see [?] in particular) and here MEGAHIT excels
350 for individual genomes even in the presence of strain variation.

351 When comparing details of sequence recovery between the assemblers,
352 the assembly content differs by only a small amount when loose alignments
353 are allowed: all three assemblers miss more content (approximately 1.8% of
354 the reference) than they generate uniquely (0.9% or less). In addition to
355 preferring no one assembler over any other, this suggests that combining as-
356 semblies may have little value in terms of recovering additional metagenome
357 content.

358 **The missing reference may be present in strain variants of the** 359 **intended species.**

360 Several individual genomes are missing in measurable portion from the QC
361 reads (Table 2), and many QC reads (4.4% of 108m) did not map to the
362 full reference metagenome. These appear to be related issues: upon analysis
363 of the unmapped reads against Genbank, we find that many of the contigs
364 assembled from the unmapped reads can be assigned to strain variants of
365 the species in the mock community (Table 9). This suggests that the con-
366 structors of the mock community may have unintentionally included strain
367 variants of *Fusobacterium nucleatum*, *Thermus thermophilus* HB27, and *En-*
368 *terococcus faecalis*. In addition, we detect what may be portions of a novel
369 member of the *Proteiniclasticum* genus in the assembly of these reads.

370 Without returning to the original DNA samples, it is impossible to con-

clusively confirm that unintended strains were used in the construction of the mock community. In particular, our analysis is dependent on the genomes in Genbank: the genomes we detect in the contigs are clearly more closely related to Genbank genomes other than the species in the reference metagenome, based on k-mer analysis and contig alignment. However, Genbank is unlikely to contain the exact genomes of the included strain variants, rendering conclusive identification impossible.

Conclusions

Overall, assembly of this mock community works well, with good recovery of known genomic sequence for the majority of genomes. All three assemblers that we evaluated recover similar amounts of most genomic sequence, but (recapitulating several other studies @cite) MEGAHIT is computationally most efficient. We note that assembly resolves substantial portions of several previously undetected strain variants, as well as recovering a substantial portion of a novel *Proteiniclasticum* spp. that was detected via amplicon analysis in [9].

The presence of closely related strains is a major confounder of metagenome assembly, and causes assemblers to drop considerable portions of genomes that (based on read mapping and k-mer inclusion) are clearly present. In this relatively simple community, this strain confusion is present but does not dominate the assembly. However, real microbial communities are likely to have many closely related strains and any resulting loss of assembly will be hard to detect in the absence of good reference genomes. While high polymorphism rates in e.g. animal genomes is known to cause duplication or loss of assembly, some solutions have emerged that make use of assumptions of uniform coverage and diploidy [29]. These solutions cannot however be transferred directly to metagenomes, which have unknown abundance distributions and strain content.

An additional concern is that metagenome assemblies are often performed after pooling data sets to increase coverage; this pooled data is more likely to contain multiple strains, which would then in turn adversely affect assembly of strains. This may not be resolvable within the current paradigm of assembly, which focuses on outputting linear assemblies that cannot properly represent strain variation. The human genomics community is moving towards using *reference graphs*, which can represent multiple incompatible variants in a single data structure [30]; this approach, however, requires high-quality isolate reference genomes, which are generally unavailable for

408 environmental metagenomes.

409 Long read sequencing (and related technologies) will undoubtedly help
410 resolve strain variation in the future, but even with highly accurate long-
411 read sequencing, current sequencing depth is still too low to resolve deep
412 environmental metagenomes [31, 32]. It is unclear how well long error-
413 prone reads (such as those output by Pacific Biosciences SMRT [33] and
414 Oxford Nanopore instruments [34]) will perform on complex metagenomes:
415 with high error rates, deep coverage of each individual genome is required
416 to achieve accurate assembly, and this may not be easily obtainable for
417 complex communities. Single-molecule barcoding (e.g. 10X Genomics [?]) and
418 HiC approaches [35] show promise but these remain untested on well-
419 defined complex communities and are still challenged by the complexity of
420 complex environmental metagenomes; see [36, 37, 38].

421 Much of our analysis depended on having a high-quality “mock” metagenome.
422 While computationally constructed synthetic communities and computa-
423 tional “spike-ins” to real data sets can provide valuable controls (e.g. see
424 [13] and [39]) we strongly believe that standardized communities constructed
425 *in vitro* and sequenced with the latest technologies are critical to the evalua-
426 tion of both canonical and emerging tools, e.g. efforts such as [40]. From the
427 perspective of tool evaluation, we must disagree somewhat with Vollmers et
428 al. [?]: good metagenome tool evaluation necessarily depends on mock
429 communities that are as realistic as we can make them. Likewise, from
430 the perspective of bench biologists, actually sequencing real DNA is critical
431 because it can evaluate confounding effects such as kit contamination [41].
432 Large-scale studies of computational approaches systematically applied to
433 mock communities such as CAMI [12] can then provide fair comparisons of
434 entire toolchains (wet + dry) applied to these mock communities.

435 We omitted two important questions in this study: binning and choice
436 of parameters. We chose not to evaluate genome binning because most
437 binning strategies either operate post-assembly (see e.g. [42]), in which case
438 the challenges with assembly discussed above will apply, or require multiple
439 samples (e.g. [43]). We also chose to use only default parameters with all
440 three assemblers, for two reasons. First, we are not aware of any widely
441 used automated approaches for determining the “best” set of parameters
442 or evaluating the output, other than those integrated into the assemblers
443 themselves (e.g. choice of k-mer sizes), and absent such guidance we do not
444 feel comfortable blessing any particular set of parameters; here the choice
445 of default parameters is parsimonious. Second, any parameter exploration
446 pipeline would not only need to be automated but would need to run multiple

447 assemblies; in this case, any comparison based on runtime of the parameter
448 choice pipeline would naturally favor MEGAHIT because of its substantial
449 advantage in computational efficiency.

450 **Author contributions**

451 SA, LI and CTB developed, tested, and executed the analytical pipeline.
452 SA and CTB created the tables and figures and wrote the paper.

453 **Competing interests**

454 No competing interest to our knowledge.

455 **Grant information**

456 This work is funded by Moore and NIH.

457 **Acknowledgments**

458 We thank Michael R. Crusoe and Phillip T. Brooks for input on analysis
459 and pipeline development.

460 **References**

- 461 [1] Jay Ghurye, Victoria Cepeda-Espinoza, and Mihai Pop. Metagenomic assem-
462 bly: Overview, challenges and applications. *The Yale Journal of Biology and*
463 *Medicine*, 89(3):353–362, 2016.
- 464 [2] Nikos C. Kyrpides, Philip Hugenholtz, Jonathan A. Eisen, Tanja Woyke,
465 Markus Göker, Charles T. Parker, Rudolf Amann, Brian J. Beck, Patrick S. G.
466 Chain, Jongsik Chun, Rita R. Colwell, Antoine Danchin, Peter Dawyndt, Tom
467 Dedeurwaerdere, Edward F. DeLong, John C. Detter, Paul De Vos, Timothy J.
468 Donohue, Xiu-Zhu Dong, Dusko S. Ehrlich, Claire Fraser, Richard Gibbs, Jack
469 Gilbert, Paul Gilna, Frank Oliver Glöckner, Janet K. Jansson, Jay D. Keasling,
470 Rob Knight, David Labeda, Alla Lapidus, Jung-Sook Lee, Wen-Jun Li, Juncai
471 MA, Victor Markowitz, Edward R. B. Moore, Mark Morrison, Folker Meyer,
472 Karen E. Nelson, Moriya Ohkuma, Christos A. Ouzounis, Norman Pace, Julian
473 Parkhill, Nan Qin, Ramon Rossello-Mora, Johannes Sikorski, David Smith,
474 Mitch Sogin, Rick Stevens, Uli Stingl, Ken ichiro Suzuki, Dorothea Taylor,
475 Jim M. Tiedje, Brian Tindall, Michael Wagner, George Weinstock, Jean Weis-
476 senbach, Owen White, Jun Wang, Lixin Zhang, Yu-Guang Zhou, Dawn Field,
477 William B. Whitman, George M. Garrity, and Hans-Peter Klenk. Genomic

478 encyclopedia of bacteria and archaea: Sequencing a myriad of type strains.
479 *PLoS Biology*, 12(8):e1001920, aug 2014. doi: 10.1371/journal.pbio.1001920.
480 URL <https://doi.org/10.1371/journal.pbio.1001920>.

481 [3] Jorge F Vázquez-Castellanos, Rodrigo García-López, Vicente Pérez-Brocal,
482 Miguel Pignatelli, and Andrés Moya. Comparison of different assembly and
483 annotation tools on analysis of simulated viral metagenomic communities in
484 the gut. *BMC genomics*, 15(1):1, 2014.

485 [4] Konstantinos Mavromatis, Natalia Ivanova, Kerrie Barry, Harris Shapiro, Eu-
486 gene Goltsman, Alice C McHardy, Isidore Rigoutsos, Asaf Salamov, Frank
487 Korzeniewski, Miriam Land, et al. Use of simulated data sets to evaluate the
488 fidelity of metagenomic processing methods. *Nature methods*, 4(6):495–500,
489 2007.

490 [5] David B Jaffe, Jonathan Butler, Sante Gnerre, Evan Mauceli, Kerstin
491 Lindblad-Toh, Jill P Mesirov, Michael C Zody, and Eric S Lander. Whole-
492 genome sequence assembly for mammalian genomes: Arachne 2. *Genome*
493 *research*, 13(1):91–96, 2003.

494 [6] Samuel Aparicio, Jarrod Chapman, Elia Stupka, Nik Putnam, Jer-ming Chia,
495 Paramvir Dehal, Alan Christoffels, Sam Rash, Shawn Hoon, Arian Smit, et al.
496 Whole-genome shotgun assembly and analysis of the genome of *fugu rubripes*.
497 *Science*, 297(5585):1301–1310, 2002.

498 [7] Anveshi Charuvaka and Huzefa Rangwala. Evaluation of short read metage-
499 nomic assembly. *BMC genomics*, 12(2):1, 2011.

500 [8] Jared T Simpson, Kim Wong, Shaun D Jackman, Jacqueline E Schein,
501 Steven JM Jones, and Inanç Birol. Abyss: a parallel assembler for short read
502 sequence data. *Genome research*, 19(6):1117–1123, 2009.

503 [9] Shakya Migun, Christopher Quince, James Campbell, Zamin Yang, Christo-
504 pher Schadt, and Mircea Podar. Comparative metagenomic and rrna microbial
505 diversity characterization using archaeal and bacterial synthetic communities.
506 *Environmental Microbiology*, 15(6):1882–1899, 2013.

507 [10] Brandon K. B. Seah and Harald R. Gruber-Vodicka. gbtools: In-
508 teractive visualization of metagenome bins in r. *Frontiers in Mi-*
509 *crobiology*, 6, dec 2015. doi: 10.3389/fmicb.2015.01451. URL
510 <https://doi.org/10.3389/fmicb.2015.01451>.

511 [11] Dinghua Li, Ruibang Luo, Chi-Man Liu, Chi-Ming Leung, Hing-
512 Fung Ting, Kunihiko Sadakane, Hiroshi Yamashita, and Tak-Wah
513 Lam. MEGAHIT v1.0: A fast and scalable metagenome assembler
514 driven by advanced methodologies and community practices. *Methods*,
515 102:3–11, jun 2016. doi: 10.1016/j.ymeth.2016.02.020. URL
516 <https://doi.org/10.1016/j.ymeth.2016.02.020>.

- [12] Alexander Sczyrba, Peter Hofmann, Peter Belmann, David Koslicki, Stefan Janssen, Johannes Droege, Ivan Gregor, Stephan Majda, Jessika Fiedler, Eik Dahms, Andreas Bremges, Adrian Fritz, Ruben Garrido-Oter, Tue Sparholt Jorgensen, Nicole Shapiro, Philip D Blood, Alexey Gurevich, Yang Bai, Dmitrij Turaev, Matthew Z DeMaere, Rayan Chikhi, Niranjan Nagarajan, Christopher Quince, Lars Hestbjerg Hansen, Soren J Sorensen, Burton K H Chia, Bertrand Denis, Jeff L Froula, Zhong Wang, Robert Egan, Dongwan Don Kang, Jeffrey J Cook, Charles Deltel, Michael Beckstette, Claire Lemaitre, Pierre Peterlongo, Guillaume Ritz, Dominique Lavenier, Yu-Wei Wu, Steven W Singer, Chirag Jain, Marc Strous, Heiner Klingenberg, Peter Meinicke, Michael Barton, Thomas Lingner, Hsin-Hung Lin, Yu-Chieh Liao, Genivaldo Gueiros Z. Silva, Daniel A Cuevas, Robert A Edwards, Surya Saha, Vitor C Piro, Bernhard Y Renard, Mihai Pop, Hans-Peter Klenk, Markus Goeker, Nikos Kyrpides, Tanja Woyke, Julia A Vorholt, Paul Schulze-Lefert, Edward M Rubin, Aaron E Darling, Thomas Rattei, and Alice C McHardy. Critical assessment of metagenome interpretation - a benchmark of computational metagenomics software. *bioRxiv*, 2017. doi: 10.1101/099127. URL <http://biorxiv.org/content/early/2017/01/09/099127>.
- [13] Andries Johannes van der Walt, Marc Warwick Van Goethem, Jean-Baptiste Ramond, Thulani Peter Makhalanyane, Oleg Reva, and Don Arthur Cowan. Assembling metagenomes, one community at a time. *bioRxiv*, 2017. doi: 10.1101/120154. URL <http://biorxiv.org/content/early/2017/06/06/120154>.
- [14] William W. Greenwald, Niels Klitgord, Victor Seguritan, Shibu Yooseph, J. Craig Venter, Chad Garner, Karen E. Nelson, and Weizhong Li. Utilization of defined microbial communities enables effective evaluation of meta-genomic assemblies. *BMC Genomics*, 18(1), apr 2017. doi: 10.1186/s12864-017-3679-5. URL <https://doi.org/10.1186/s12864-017-3679-5>.
- [15] Yu Peng, Henry C.M. Leung, S.M. Yiu, and Francis Y.L. Chin. Idba-ud: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics*, 28:1420–1428, 2012.
- [16] Sergey Nurk, Dmitry Meleshko, Anton Korobeynikov, and Pavel A. Pevzner. metaSPAdes: a new versatile metagenomic assembler. *Genome Research*, 27(5):824–834, mar 2017. doi: 10.1101/gr.213959.116. URL <https://doi.org/10.1101/gr.213959.116>.
- [17] Dinghua Li, Ruibang Luo, Chi-Man Liu, Chi-Ming Leung, Hing-Fung Ting, Kunihiko Sadakane, Hiroshi Yamashita, and Tak-Wah Lam. Megahit v1.0: A fast and scalable metagenome assembler driven by advanced methodologies and community practices. *Methods*, 102:3–11, 2016.
- [18] H Chitsaz, JL Yee-Greenbaum, G Tesler, MJ Lombardo, CL Dupont, JH Badger, M Novotny, DB Rusch, LJ Fraser, NA Gormley, O Schulz-Trieglaff,

GP Smith, DJ Evers, PA Pevzner, and RS Lasken. Efficient de novo assembly of single-cell bacterial genomes from short-read data sets. *Nat Biotechnol*, 29(10):915–21, 2011.

[19] Anthony M. Bolger, Marc Lohse, and Bjoern Usadel. Trimmomatic: A flexible trimmer for illumina sequence data. *Bioinformatics*, 30(15):2114–2120, 2014.

[20] Matthew D MacManes. On the optimal trimming of high-throughput mrna sequence data. *Frontiers in genetics*, 5:13, 2014.

[21] Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[22] C. Titus Brown and Luiz Irber. sourmash: a library for MinHash sketching of DNA. *The Journal of Open Source Software*, 1(5), sep 2016. doi: 10.21105/joss.00027. URL <https://doi.org/10.21105/joss.00027>.

[23] Brian D. Ondov, Todd J. Treangen, Páll Melsted, Adam B. Mallonee, Nicholas H. Bergman, Sergey Koren, and Adam M. Phillippy. Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biology*, 17(1), jun 2016. doi: 10.1186/s13059-016-0997-x. URL <https://doi.org/10.1186/s13059-016-0997-x>.

[24] David Koslicki and Daniel Falush. Metapalette: a k-mer painting approach for metagenomic taxonomic profiling and quantification of novel strain variation. *mSystems*, 1(3), 2016. doi: 10.1128/mSystems.00020-16. URL <http://msystems.asm.org/content/1/3/e00020-16>.

[25] Zhang Qingpeng, Awad Sherine, and Brown Titus. Crossing the streams: a framework for streaming analysis of short dna sequencing reads. *PeerJ PrePrints 3:e1100* <https://dx.doi.org/10.7287/peerj.preprints.890v1>, 2015.

[26] MR Crusoe, HF Alameldin, S Awad, E Boucher, A Caldwell, R Cartwright, A Charbonneau, B Constantinides, G Edverson, S Fay, J Fenton, T Fenzl, J Fish, L Garcia-Gutierrez, P Garland, J Gluck, I Gonzlez, S Guermond, J Guo, A Gupta, JR Herr, A Howe, A Hyer, A Hrpfer, L Irber, R Kidd, D Lin, J Lippi, T Mansour, P McA’Nulty, E McDonald, J Mizzi, KD Murray, JR Nahum, K Nanlohy, AJ Nederbragt, H Ortiz-Zuazaga, J Ory, J Pell, C Pepe-Ranne, ZN Russ, E Schwarz, C Scott, J Seaman, S Sievert, J Simpson, CT Skennerton, J Spencer, R Srinivasan, D Standage, JA Stapleton, SR Steinman, J Stein, B Taylor, W Trimble, HL Wiencko, M Wright, B Wyss, Q Zhang, e zyme, and CT Brown. The khmer software package: enabling efficient nucleotide sequence analysis [version 1; referees: 2 approved, 1 approved with reservations]. *F1000Research*, 4(900), 2015. doi: 10.12688/f1000research.6924.1.

- [27] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, et al. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–2079, 2009.
- [28] Stefan Kurtz, Adam Phillippy, Arthur L Delcher, Michael Smoot, Martin Shumway, Corina Antonescu, and Steven L Salzberg. Versatile and open software for comparing large genomes. *Genome biology*, 5(2):1, 2004.
- [29] J. H. Kim, M. S. Waterman, and L. M. Li. Diploid genome reconstruction of *Ciona intestinalis* and comparative analysis with *Ciona savignyi*. *Genome Research*, 17(7):1101–1110, jun 2007. doi: 10.1101/gr.5894107. URL <https://doi.org/10.1101/gr.5894107>.
- [30] Benedict Paten, Adam M Novak, Jordan M Eizenga, and Erik Garrison. Genome graphs and the evolution of genome inference. *Genome research*, 27(5):665–676, 2017.
- [31] Itai Sharon, Michael Kertesz, Laura A. Hug, Dmitry Pushkarev, Timothy A. Blauwkamp, Cindy J. Castelle, Mojgan Amirebrahimi, Brian C. Thomas, David Burstein, Susannah G. Tringe, Kenneth H. Williams, and Jillian F. Banfield. Accurate, multi-kb reads resolve complex populations and detect rare microorganisms. *Genome Research*, 25(4):534–543, feb 2015. doi: 10.1101/gr.183012.114. URL <https://doi.org/10.1101/gr.183012.114>.
- [32] Richard Allen White, Eric M. Bottos, Taniya Roy Chowdhury, Jeremy D. Zucker, Colin J. Brislawn, Carrie D. Nicora, Sarah J. Fansler, Kurt R. Glaesemann, Kevin Glass, and Janet K. Jansson. Molecule long-read sequencing facilitates assembly and genomic binning from complex soil metagenomes. *mSystems*, 1(3):e00045–16, jun 2016. doi: 10.1128/msystems.00045-16. URL <https://doi.org/10.1128/msystems.00045-16>.
- [33] J. Eid, A. Fehr, J. Gray, K. Luong, J. Lyle, G. Otto, P. Peluso, D. Rank, P. Baybayan, B. Bettman, A. Bibillo, K. Bjornson, B. Chaudhuri, F. Christians, R. Cicero, S. Clark, R. Dalal, A. deWinter, J. Dixon, M. Foquet, A. Gaertner, P. Hardenbol, C. Heiner, K. Hester, D. Holden, G. Kearns, X. Kong, R. Kuse, Y. Lacroix, S. Lin, P. Lundquist, C. Ma, P. Marks, M. Maxham, D. Murphy, I. Park, T. Pham, M. Phillips, J. Roy, R. Sebra, G. Shen, J. Sorenson, A. Tomaney, K. Travers, M. Trulson, J. Vieceli, J. Wegener, D. Wu, A. Yang, D. Zaccarin, P. Zhao, F. Zhong, J. Korlach, and S. Turner. Real-time DNA sequencing from single polymerase molecules. *Science*, 323(5910):133–138, jan 2009. doi: 10.1126/science.1162986. URL <https://doi.org/10.1126/science.1162986>.
- [34] Gerald M Cherf, Kate R Lieberman, Hytham Rashid, Christopher E Lam, Kevin Karplus, and Mark Akeson. Automated forward and reverse ratcheting of DNA in a nanopore at 5-AA precision. *Nature Biotechnology*, 30(4):344–348, feb 2012. doi: 10.1038/nbt.2147. URL <https://doi.org/10.1038/nbt.2147>.

- [35] Caiti Smukowski Heil, Joshua N. Burton, Ivan Liachko, Anne Friedrich, Noah A. Hanson, Cody L. Morris, Joseph Schacherer, Jay Shendure, James H. Thomas, and Maitreya J. Dunham. Identification of a novel interspecific hybrid yeast from a metagenomic open fermentation sample using hi-c. *bioRxiv*, 2017. doi: 10.1101/150722. URL <http://biorxiv.org/content/early/2017/06/15/150722>.
- [36] Joshua N. Burton, Ivan Liachko, Maitreya J. Dunham, and Jay Shendure. Species-level deconvolution of metagenome assemblies with hi-c-based contact probability maps. *G3*, 4(7):1339–1346, may 2014. doi: 10.1534/g3.114.011825. URL <https://doi.org/10.1534/g3.114.011825>.
- [37] Martial Marbouty, Axel Cournac, Jean-François Flot, Hervé Marie-Nelly, Julien Mozziconacci, and Romain Koszul. Metagenomic chromosome conformation capture (meta3c) unveils the diversity of chromosome organization in microorganisms. *eLife*, 3, dec 2014. doi: 10.7554/elife.03318. URL <https://doi.org/10.7554/elife.03318>.
- [38] Christopher W. Beitel, Lutz Froenicke, Jenna M. Lang, Ian F. Korf, Richard W. Micheltore, Jonathan A. Eisen, and Aaron E. Darling. Strain- and plasmid-level deconvolution of a synthetic metagenome by sequencing proximity ligation products. *PeerJ*, 2:e415, may 2014. doi: 10.7717/peerj.415. URL <https://doi.org/10.7717/peerj.415>.
- [39] Adina Chuang Howe, Janet K Jansson, Stephanie A Malfatti, Susannah G Tringe, James M Tiedje, and C Titus Brown. Tackling soil diversity with the assembly of large, complex metagenomes. *Proceedings of the National Academy of Sciences*, 111(13):4904–4909, 2014.
- [40] Bonnie L. Brown, Mick Watson, Samuel S. Minot, Maria C. Rivera, and Rima B. Franklin. MinIONTM nanopore sequencing of environmental metagenomes: a synthetic approach. *GigaScience*, 6(3):1–10, feb 2017. doi: 10.1093/gigascience/gix007. URL <https://doi.org/10.1093/gigascience/gix007>.
- [41] Susannah J Salter, Michael J Cox, Elena M Turek, Szymon T Calus, William O Cookson, Miriam F Moffatt, Paul Turner, Julian Parkhill, Nicholas J Loman, and Alan W Walker. Reagent and laboratory contamination can critically impact sequence-based microbiome analyses. *BMC Biology*, 12(1), nov 2014. doi: 10.1186/s12915-014-0087-z. URL <https://doi.org/10.1186/s12915-014-0087-z>.
- [42] Cedric C Laczny, Christina Kiefer, Valentina Galata, Tobias Fehlmann, Christina Backes, and Andreas Keller. Busybee web: metagenomic data analysis by bootstrapped supervised binning and annotation. *Nucleic Acids Research*, page gkx348, 2017.

674 [43] Brian Cleary, Ilana Lauren Brito, Katherine Huang, Dirk Gevers, Terrance
675 Shea, Sarah Young, and Eric J Alm. Detection of low-abundance bacte-
676 rial strains in metagenomic datasets by eigengenome partitioning. *Nature*
677 *Biotechnology*, 33(10):1053–1060, sep 2015. doi: 10.1038/nbt.3329. URL
678 <https://doi.org/10.1038/nbt.3329>.