Assembling reads improves taxonomic classification of species

Quang Tran* and Vinhthuy Phan

*Correspondence:
qmtran@memphis.edu
Department of Computer Science,
University of Memphis, Memphis,
Tennessee 38152, USA
Full list of author information is available at the end of the article

Abstract

Background: Most current metagenomic classifiers and profilers employ short reads to classify, bin and profile microbial genomes that are present in metagenomic samples. Many of these methods adopt techniques that aim to identify unique genomic regions of genomes so as to differentiate them. Because of this, short-read lengths might be suboptimal. Longer read lengths might improve the performance of classification and profiling. However, longer reads produced by current technology tend to have a higher rate of sequencing errors, compared to short reads. It is not clear if the trade-off between longer length versus higher sequencing errors will increase or decrease classification and profiling performance.

Results: We compared performance of popular metagenomic classifiers on short reads and longer reads, which are assembled from the same short reads. When using a number of popular assemblers to assemble long reads from the short reads, we discovered that most classifiers made fewer predictions with longer reads and that they achieved higher classification performance on synthetic metagenomic data. Specifically, across most classifiers, we observed a significant increase in precision, while recall remained the same, resulting in higher overall classification performance. On real metagenomic data, we observed a similar trend that classifiers made fewer predictions. This suggested that they might have the same performance characteristics of having higher precision while maintaining the same recall with longer reads.

Conclusions: This finding has two main implications. First, it suggests that classifying species in metagenomic environments can be achieved with higher overall performance simply by assembling short reads. This suggested that they might have the same performance characteristics of having higher precision while maintaining the same recall as shorter reads. Second, this finding suggests that it might be a good idea to consider utilizing long-read technologies in species classification for metagenomic applications. Current long-read technologies tend to have higher sequencing errors and are more expensive compared to short-read technologies. The trade-offs between the pros and cons should be investigated.

Keywords: metagenomic classification; short-read sequencing; metagenomic assembly

1 Background

Species classification and profiling is an important problem in metagenomics analysis. Many different computational tasks and workflows depend on the identification and profiling of organisms that are present in metagenomics samples. Examples include studies that assessed the host-microbe interactions in the gut microbiome to gain better insight into human health [1], revealed ecological differentiation of
closely related bacteria [2], uncovered the presence of ancient sub-populations of marine bacteria [3], and highlighted extensive intra-species recombination [4, 5].

Methods for classification and profiling of microbial communities are diverse. CLARK [6] uses a database of k-mers that aims to uniquely describe genomic regions of each targeted microbes. GOTTCHA [7] has a different approach to identifying unique genomic regions of targeted microbes by using a combination of empirical data and machine learning methods. Kraken [8] also utilizes k-mers, but builds taxonomic trees that help differentiate closely related microbes. MetaPhlAn2 [9] employs a similar taxonomic approach, but narrows read alignment and its analysis only on a set of around one million markers.

Although short reads have low sequencing errors, their short lengths have been a limitation in the identification of structural variants, sequencing repetitive regions, phasing of alleles and distinguishing highly homologous genomic regions. This limitation can have a serious practical consequence. For example, it might have significantly contributed to the diagnostic gap in patients with genetic disorders [10].

More recent technologies can produce very long reads, but at the expense of having higher costs and much higher error rates [11]. However, longer reads have been found to be more appropriate or better compared to short reads in certain studies. Single-molecule sequencing (SMS) offers exceptionally long reads that enable direct sequencing of genomic regions that are difficult to sequence with short reads, including long repetitive elements, extreme GC-content regions, and complex gene loci. Similarly, these platforms enable structural variation characterization at previously unparalleled resolution and direct detection of epigenetic marks in native DNA [12]. Similarly, the PacBio sequencing system can capture full-length 16S rRNA sequences [13]. Third-generation nanopore sequencing offers many solutions to the current problems of using whole metagenome sequencing (WMS) for infectious disease diagnostics. It has been successfully utilized for pathogen detection, AMR prediction, and characterization of mixed microbial communities [14].

While long read technologies are more appropriate for certain studies, short read technologies are mature and less expensive. Is it possible to leverage known strengths of short read technologies to garner the high performance of long reads?

In this paper, we demonstrate that it is possible to improve the performance of species classification in metagenomic applications using long reads that are assembled from short reads. This finding has two major implications. First, it suggests that many existing studies that utilize short reads can benefit from long reads that are assembled from the short reads. Although there is an extra computational cost of assembly and minor modification to the existing workflows, the increase in performance might justify the cost. Second, this finding suggests that there are potential gains in utilizing long-reads technologies in this type of applications. As current long-read technologies have different characteristics from short-read technologies in terms of cost and sequencing errors, the trade-offs between these pros and cons remain to be investigated.

2 Results
2.1 Experimental Design
The main hypothesis is that the classification or identification of microbes in metagenomics samples is better done with long reads than with short reads. We aim to de-
sign a controlled experiment to verify the hypothesis. To achieve this, we evaluated the ability to detect species in metagenomics samples of several well-known classifiers on several short-reads datasets and derived long-reads datasets. The choice of which long-reads datasets are used to compare against which short-reads datasets is an important design decision. If we choose a long-reads dataset produced by a current technology to compare against a short-reads dataset produced by a different technology, the result might be due to differences in technologies rather than in read lengths. As our goal is to examine the impact of read lengths on classification, we chose to use long reads that are derived from the same short reads. These derived long-reads datasets are constructed by assembling short reads from the datasets that are used to evaluate the classifiers’ performance. Although this design choice removes the effect of sequencing technologies, it introduces the potential effect of assembling reads on the result. To address this, we evaluated classifiers with different assemblers to remove algorithmic bias on classification performance.

More descriptions of the experimental design can be found in section Methods.

2.2 Performance Assessment
Classifiers were evaluated with synthetic and real samples. Although some tools can work on the strain level, we evaluate classification results at species levels since most methods still do not provide strain level identifications. Classification performance of synthetic data was measured in terms of precision, recall, and F1.

Precision = \( \frac{TP}{TP + FP} \); Recall = \( \frac{TP}{TP + FN} \); F1 = \( \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \)

Where TP (true positives): the number of correctly classified species, FP (false positives): the number of incorrectly classified species, FN (false negatives): the number of incorrectly classified non-species, by each method.

To access classifiers’ performance on real data, we define the overall pairwise similarity of a method c to other methods as

\[
\sum_{i=1, i \neq c}^{n} \frac{|S_c \cap S_i|}{\sum_{i=1}^{n} |S_c \cap S_i|}
\]

where, \( S_i \) is the number of species predicted by method \( i \). This similarity is between 0 and 1. The closer it is to 1, the higher the overall similarity to other methods.

2.2.1 Data
Mende datasets [15] were simulated for Sanger sequencing, pyrosequencing, and Illumina sequencing. For each technology, three metagenomes were simulated to mimic different community complexities 10 species (10s), 100 species (100s), and 400 species (400s). However, the Sanger sequencing, pyrosequencing technologies seem absolute/out-of-date. We test our hypothesis on Illumina paired-end raw reads of Mende datasets, which is a very widely used sequencing platform.

To test our hypothesis with the real data, we used the gut microbiome data [16]. The metagenomic shotgun-sequencing data for two samples (ERR2017411,
ERR2017412) was downloaded from the European Bioinformatics Institute (EBI) database under the accession code ERP023788.

All data used was paired-end reads within the Illumina platform. Synthetic data consists of 26 million reads for each dataset (10s, 100s, and 400s) with the read length of 75bp. Real data has 17 million reads for each sample with the read length of 90bp. There is a slightly different in read lengths between synthetic data and real data; however, the read lengths from 75bp to 100bp have been reported [17, 18] to produce the same alignment results.

2.3 Findings
Using assembled reads, four out of seven classifiers increased their precision by at most double, while maintaining similar recall; see Table 1. These four classifiers are Kaiju, CLARK, Kraken and MetaCache. The improvement in performance was most drastic for smaller datasets. With the dataset 10s, which consisted of 10 species, CLARK, for example, benefited from a 50x increase in precision with the same recall, when reads were assembled by any of the three assemblers. With the dataset 100s, which consisted of 100 species, CLARK benefited from 3x-4x increase in precision with the same recall, when reads were assembled by any of the three assemblers. With the dataset 400s, which consisted of 400 species, CLARK benefited from a 1.04x increase in precision with the same recall. Similarly, other three classifiers benefited from assembled reads. Kraken and MetaCache benefited from increases in both precision and recall with the larger datasets 100s and 400s.

MetaPhlAn2’s performance got worse with assembled reads, compared to its performance on unassembled short reads. As seen in Figure 2, the overall F1 score is highest when reads were not assembled. We also observed that DUDes and GOTTCHA did not benefit from assembled reads. Figure 2 shows that the F1 scores of DUDes and GOTTCHA remained unchanged or decreased slightly with assembled reads. We also observed that F1 scores were highest when reads were assembled by MEGAHIT and metaSPAdes.

We observed that with synthetic, the majority of classifiers predicted much fewer species when reads were assembled; see Table 3. This observation likely explains the drastic observed increase in precision, while maintaining similar recall, across the board with synthetic datasets. With real datasets, we observed a similar behavior that classifiers predicted much fewer species when reads were assembled. Although we could not compute precision and recall with real data, the same trend suggests that as with synthetic data, classifiers should be much more precise when reads were assembled. This increase in precision should, similarly, be drastic when the datasets have much fewer species than the index database that were used by classifiers to classify species.

Additionally, Table 4 shows that with real datasets, the overall pairwise similarity decreased with assembled reads. This suggests that with assembled reads, classifiers had a higher chance of showing their uniqueness in predicting species.

2.4 Discussion
In this work, we showed the promising prospect of utilizing long reads in identifying species in metagenomic samples. Long reads, used in this study, are assembled from
the same short reads, which were used to compare classification performance. This was performed to remove potential side effects of different sequencing technologies. As future long-read technologies achieve fewer sequencing errors and become less expensive, their use for species classification in metagenomics should be desirable.

At presence, we have demonstrated that we can leverage the advantage of long reads by assembling short reads that would otherwise be used for species classification. We showed that at least two of the currently popular assemblers can be used for this purpose. We observed that MEGAHIT and metaSPades produced higher N50s across datasets, while Ray had lower N50s. In fact, it failed to assemble reads when the datasets contained 400 species. An quick comparison between metaSPAdes and MEGAHIT assemblers across all the datasets considered in this study confirmed that metaSPAdes performs better for a smaller dataset (10s) while MEGAHIT performs better for larger datasets (100s and 400s).

We think that Kaiju, CLARK, Kraken, and MetaCache benefited from the longer reads because their approach of k-mers as unique markers to distinguish closely related species. On the other hand, MetaPhlAn2, DUDes and GOTTCHA have built-in statistical post-processing procedures that align reads to reference genomes, which appear not benefit from longer reads.

3 Methods
Most metagenomic classifiers, including those that we studied in this paper, consist of two main steps. In the preprocessing step, a classifier utilizes reference genomic sequence of existing species to build an index or reference table. The index or reference table is built only once for a metagenomic environment. In the classification step, the classifier uses the index or reference table to classify metagenomic data, in the form of reads, and make predictions.

Our method interrupts this workflow by modifying the classification step. Before feeding short reads as inputs to a classifier, we assemble them into long reads. Figure 1 depicts the process of comparing a classifier’s performance on short reads and long reads. Figure 1A is the standard workflow of a classifier, in which the classifier takes as input a short reads dataset and outputs species that it predicts to be present in the sample. Figure 1B shows a workflow, in which the same short reads are first assembled before feeding to the classifier.

Different methods may have different types of prediction outputs, which can be species label for each read, or predicted species for the entire dataset, or predicted percentages of species in the sample (in case of metagenomics profiling). Profiling classifiers output a rank separated taxonomic profile with relative abundances, whereas binning classifiers provide sequence identification, length used in the assignment, and taxon as output. These outputs are then converted into species names. As a result, each metagenomic classifier produces a list of species as output.

Each classifier requires a reference database of genomic sequences to classify metagenomic reads into species. We used complete genomes of bacteria archaea, and viruses from NCBI to construct this database for each classifier. We removed species that were labeled unclassified or unknown because they might cause problems for taxonomic prediction [19].
For consistency, we used the NCBI taxonomy database [20] to standardize results from different classifiers. Further, for classifiers that produced strain-level predictions, we converted them to species-level predictions so that the results can be compared consistently across different classifiers.

We compared the outputs of classifiers using default parameters at the species level because not all classifiers still predicted at strain level. Species is a taxonomic rank more relevant in clinical diagnostics or pathogen identification than genus or phylum. Although some clinical diagnosis and epidemiological tracking often requires identification of strains, genomic databases remain poorly populated below the species level [21]. Evaluation was done in a similar way to [21, 22]. For each classifier, we evaluated predicted species produced with assembled reads and predicted species produced with original short reads.

3.1 Classifiers
We evaluated with a set of seven metagenomic classifiers: Kaiju (version 1.7.2) [23], CLARK (version 1.2.6) [6], Kraken (version 1.1.1) [8], MetaCache (version 0.6.1) [24], MetaPhlAn2 (version 2.6.0) [9], DUDes (version 0.08) [25], and GOTTCHA (version 1.0c) [7]. The choice was motivated by recent publications comparing the performance of such tools [21].

Kaiju, CLARK, Kraken, MetaCache are k-mer based methods for metagenomic read classification. CLARK and Kraken were run with the default k-mer size of 31, while MetaCache use 16-mers by default. Kaiju was run in the fastest MEM mode (with minimum fragment length \(m = 11\)), as well as in the heuristic greedy mode (with minimum score \(s = 65\)).

On the other hand, both MetaPhlAn2 [9] and DUDes [25] have to use results of read-to-reference mapping from Bowtie2 [26]; however, for some longer contigs (several million bps), Bowtie2 (version 2.3.4.2) crashed. We used a read mapper designed for both short and long reads: Minimap2 (version 2.17) [27] as an alternative for mapping reads to reference genomes.

While running the classifiers above, we specified the “paired-end reads” option for raw read input as well as the “singleton read” option for assembled read input.

3.2 Assemblers
MEGAHIT (version 1.2.9) [28], metaSPAdes (version 3.13.1) [29], Ray (version 2.3.1) [30] have been used to assemble short-reads into contigs. These tools were selected based on their popularity for assembling metagenomic reads [31].

Assemblers have been launched with (mostly) default parameters; taking a pair of FASTQ files that contains raw reads and then producing a single FASTA file that contains assembled reads for each dataset. The file names of assembled reads from MEGAHIIT, metaSPAdes, and Ray are “final.contigs”, “contigs” and “Contigs” respectively.

Ray parallelizes assembly computations using the Message Passing Interface (MPI) standard, a run agent “napirun”. While metaSPAdes consumes very high memory, MEGAHIIT specifies multiple computational threads and optionally a graphical processing unit for improving its runtime. Due to the scope of this work, we do not report the runtime as well as memory usage of the assemblers.
4 Conclusions

We compared performance of popular metagenomic classifiers on short reads and longer reads, which are assembled from the same short reads. Using a number of popular assemblers to assemble short reads, we discovered that most classifiers made fewer predictions with longer reads and that they achieved higher classification performance on synthetic metagenomic data. Specifically, across most classifiers, we observed a significant increase in precision, while recall remained the same, resulting in higher overall classification performance. On real metagenomic data, we observed a similar trend as in the case of synthetic data that classifiers made fewer predictions. This suggested that they might have the same performance characteristics of having higher precision while maintaining the same recall with longer reads.

This finding has two main implications. First, it suggests that classifying species in metagenomic environments can be achieved with higher overall performance simply by assembling short reads. This finding can make a big impact on the many existing studies that utilize short reads. The modification to their existing workflow is minimal, although there is an extra computational cost of assembling short reads. We showed that a number of existing assemblers could be used for the purpose of assembling short reads into longer contigs for this specific purpose.

Second, this finding suggests that it might be a good idea to consider utilizing long-read technologies in species classification for metagenomic applications. Current long-read technologies tend to have higher sequencing errors and are more expensive compared to short-read technologies. The trade-offs between the pros and cons should be investigated.

Abbreviations
SMS: Single-molecule sequencing; GC: Guanine-cytosine; PacBio: Pacific Biosciences; rRNA: Ribosomal ribonucleic acid; WMS: Whole metagenome sequencing; AMR: Antimicrobial resistance; NCBI: National Center for Biotechnology Information; MPI: Message Passing Interface

Declarations

Ethics approval and consent to participate
This manuscript does not report data collected from humans or animals.

Consent for publication
This manuscript does not contain any individual person’s data in any form.

Availability of data and material
All datasets analyzed during the current study were downloaded from public databases which were cited within the manuscript. All scripts implementing the described analyses are available at https://github.com/Coaxecva/AssembledReadsEval.

Competing interests
The authors declare that they have no competing interests.

Funding
QT was funded by a University of Memphis teaching assistantship. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript

Author’s contributions
Conceived and designed the experiments: QT, VP. Performed the experiments: QT. Analyzed the data: QT, VP. Contributed reagents/materials/analysis tools: QT. Wrote the paper: QT, VP.

Acknowledgements
We are grateful for the fruitful discussions with Vitor Piro (Robert Koch Institute), Rachid Ounit (University of California, Riverside). We’d also like to thank Vinod Gupta (Mayo Clinic) for sharing data, useful suggestions, and Eric Spangler (University of Memphis) for research high performance computing support.
References

1. Bonder, M.J., Kurilshikov, A., Tigchelaar, E.F., Mujagic, Z., Imhann, F., Vila, A.V., Deelen, P., Vatanen, T., Schirmer, M., Smeekens, S.P., et al.: The effect of host genetics on the gut microbiome. Nature genetics 48(11), 1407–1412 (2016)

2. Shapiro, B.J., Friedman, J., Cordero, O.X., Preheim, S.P., Timberlake, S.C., Szabó, G., Polz, M.F., Alm, E.J.: Population genomics of early events in the ecological differentiation of bacteria. science 336(6077), 48–51 (2012)

3. Kaashtan, N., Roggenack, S.E., Rodrigue, S., Thompson, J.W., Biller, S.J., Cole, A., Ding, H., Martinen, P., Malmstrom, R.R., Stocker, R., et al.: Single-cell genomics reveals hundreds of coexisting subpopulations in wild prochlorococcus. Science 344(6182), 416–420 (2014)

4. Snitkin, E.S., Zalazny, A.M., Montero, C.I., Stock, F., Mijares, L., Murray, P.R., Segre, J.A., et al.: Genome-wide recombination drives diversification of epidemic strains of acinetobacter baumannii. Proceedings of the National Academy of Sciences 108(33), 13758–13763 (2011)

5. Rosen, M.J., Davison, M., Bhaya, D., Fisher, D.S.: Fine-scale diversity and extensive recombination in a quasisexual bacterial population occupying a broad niche. Science 340(6182), 1019–1023 (2013)

6. Ounit, R., Wanamaker, S., Close, T.J., Lonardi, S.: Clark: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. BMC genomics 16(1), 236 (2015)

7. Freitas, T.A.K., Li, P.-E., Scholz, M.B., Chain, P.S.: Accurate read-based metagenome characterization using a hierarchical suite of unique signatures. Nucleic acids research 43(10), 68–69 (2015)

8. Wood, D.E., Salzberg, S.L.: Kraken: ultrafast metagenomic sequence classification using exact alignments. Genome biology 15(3), 46 (2014)

9. Truong, D.T., Franzosa, E.A., Tickle, T.L., Scholz, M., Weingtart, G., Pasolli, E., Tett, A., Hutenhower, C., Segata, N.: Metaphlan2 for enhanced metagenomic taxonomic profiling. Nature methods 12(10), 902 (2015)

10. Mantere, T., Kersten, S., Hoischen, A.: Long-read sequencing emerging in medical genetics. Frontiers in genetics 10, 426 (2019)

11. Lau, B., Mohiyudin, M., Mu, J.C., Fang, L.T., Bani Asadi, N., Dallett, C., Lam, H.Y.: Longslnd: in silico sequencing of lengthy and noisy datatypes. Bioinformatics 32(24), 3829–3832 (2016)

12. Amer, A., Kloosterman, W.P., Hestand, M.S.: Single-molecule sequencing: towards clinical applications. Trends in biotechnology 37(1), 72–85 (2019)

13. Pootakham, W., Mhuantong, W., Yoocha, T., Putchim, L., Sonthirod, C., Naktang, C., Thongtham, N., Tangphatsornruang, S.: High resolution profiling of coral-associated bacterial communities using full-length 16s rna sequence data from pacbio smrt sequencing system. Scientific reports 9, 1–12 (2019)

14. Petersen, L.M., Martin, I.W., Moschetti, W.E., Kershaw, C.M., Tsongalis, G.J.: Third-generation sequencing in the clinical laboratory: Exploring the advantages and challenges of nanopore sequencing. Journal of Clinical Microbiology 58(1) (2019)

15. Mende, D.R., Waller, A.S., Sunagawa, S., Järvelin, A.I., Chan, M.M., Arumugam, M., Raes, J., Bork, P.: Assessment of metagenomic assembly using simulated next generation sequencing data. PLoS one 7(2), 31386 (2012)

16. Jie, Z., Xia, H., Zhong, S.L., Feng, Q., Li, S., Liang, S., Zhong, H., Liu, Z., Gao, Y., Zhao, H., et al.: The gut microbiome in atherosclerotic cardiovascular disease. Nature communications 7(1), 1–12 (2016)

17. Phan, V., Gao, S., Tran, Q., Vo, N.S.: How genome complexity can explain the hardness of aligning reads to genomes. In: 2014 IEEE 4th International Conference on Computational Advances in Bio and Medical Sciences (ICCABS), pp. 1–2 (2014). IEEE

18. Tran, Q., Gao, S., Vo, N.S., Phan, V.: A linear model for predicting performance of short-read aligners using genome complexity. BMC bioinformatics 16(15), 17 (2015)

19. Wooley, J.C., Godzik, A., Friedberg, I.: A primer on metagenomics. PLoS Comput Biol 6(2), 1000667 (2010)

20. Malmstrom, R.R., Stocker, R., Segata, N.: Comprehensive benchmarking and ensemble approaches for metagenomic classifiers. Genome biology 18(1), 182 (2017)

21. Menzel, P., Ng, K.L., Krogh, A.: Fast and sensitive taxonomic classification for metagenomics with kaiju. Nature communications 7(1), 1–9 (2016)

22. Müllner, A., Hundt, C., Hildebrandt, A., Hankeln, T., Schmidt, B.: Metacache: context-aware classification of complex metagenomics assembly via succinct de bruijn graph. Bioinformatics 31(10), 1674–1676 (2015)

23. Menzel, P., Ng, K.L., Krogh, A.: Fast and sensitive taxonomic classification for metagenomics with kaiju. Nature communications 7(1), 1–9 (2016)

24. Müllner, A., Hundt, C., Hildebrandt, A., Hankeln, T., Schmidt, B.: Metacache: context-aware classification of complex metagenomics assembly via succinct de bruijn graph. Bioinformatics 31(10), 1674–1676 (2015)

25. Nurk, S., Meleshko, D., Korobeynikov, A., Pevzner, P.A.: metaspades: a new versatile metagenomic assembler. Genome research 27(5), 824–834 (2017)

26. Boisvert, S., Raymond, F., Godzariadis, É., Laviolette, F., Corbell, J.: Ray meta: scalable de novo metagenome assembly and profiling. Genome biology 13(12), 122 (2012)

27. Aylings, M., Clark, M.D., Leggett, R.M.: New approaches for metagenome assembly with short reads. Briefings in bioinformatics (2019)
### Table 1

Precision, recall, F-1 of species-level classification of four metagenomic classifiers on three synthetic short read datasets, which are, respectively, not assembled and assembled by three assemblers.

|          | Kaiju Pre | Kaiju Rec | Kaiju F1 | CLARK Pre | CLARK Rec | CLARK F1 | Kraken Pre | Kraken Rec | Kraken F1 | MetaCache Pre | MetaCache Rec | MetaCache F1 |
|----------|------------|-----------|----------|-----------|-----------|----------|------------|------------|----------|---------------|---------------|--------------|
| 10s      | .02        | 1.0       | .04      | .02       | 1.0       | .05      | .03        | 1.0        | .06      | .20           | 1.0           | .33          |
| not assembled | MEGAHIT | .50        | .90      | .64      | 1.0       | 1.0      | 1.0        | 1.0        | 1.0      | .90           | 1.0           | .95          |
|          | MetaSPAdes | .50        | .90      | .64      | 1.0       | 1.0      | 1.0        | 1.0        | 1.0      | .66           | 1.0           | .80          |
|          | Ray        | .39        | .90      | .54      | 1.0       | 1.0      | 1.0        | 1.0        | 1.0      | .83           | 1.0           | .91          |
| 100s     | .18        | .87        | .29      | .21       | .98       | .35      | .21        | .84        | .34      | .47           | .97           | .63          |
| not assembled | MEGAHIT | .35        | .87      | .50      | .88       | .99      | .93        | .67        | .86      | .75           | .78           | .87          |
|          | MetaSPAdes | .35        | .87      | .50      | .69       | .99      | .81        | .63        | .86      | .73           | .73           | .99          |
|          | Ray        | .25        | .87      | .38      | .98       | .99      | .98        | .75        | .86      | .80           | .83           | .99          |
| 400s     | .84        | .88        | .86      | .95       | .99       | .97      | .95        | .83        | .88      | .91           | .97           | .94          |
| not assembled | MEGAHIT | .88        | .88      | .88      | .99       | .99      | .99        | .98        | .84      | .90           | .97           | .99          |
|          | MetaSPAdes | .87        | .88      | .87      | .98       | .99      | .99        | .98        | .84      | .90           | .96           | .99          |
|          | Ray        | .95        | .85      | .90      | .99       | .99      | .99        | .99        | .84      | .91           | .99           | .98          |
Table 2 Assembly statistics for all assemblers on simulated (10s, 100s, 400s) and real (ERR2017411, ERR2017412) data

| Statistics                  | Dataset        | MEGAHIT | metaSPAdes | Ray   |
|-----------------------------|---------------|---------|------------|-------|
|                             | Synthetic Data|         |            |       |
| number of contigs           | 10s           | 1,069   | 1,156      | 3,256 |
| largest contig              | 10s           | 835,563 | 1,436,250  | 294,361 |
| avg contig                  | 10s           | 31,529.53 | 29,211.97 | 10,307.24 |
| n50                         | 10s           | 131,416 | 234,206    | 31,735 |
| number of contigs           | 100s          | 156,074 | 218,705    | 717,512 |
| largest contig              | 100s          | 573,139 | 190,202    | 14,995 |
| avg contig                  | 100s          | 1,936.78 | 1,448.98  | 189.24 |
| n50                         | 100s          | 3,051   | 2,732      | 177   |
| number of contigs           | 400s          | 488,142 | 901,182    | 59,663 |
| largest contig              | 400s          | 21,914  | 13,618     | 3,367 |
| avg contig                  | 400s          | 377.24  | 323.58     | 149.72 |
| n50                         | 400s          | 361     | 319        | 138   |
|                             | Real Data     |         |            |       |
| number of contigs           | ERR2017411    | 85,426  | 165,252    | 252,974 |
| largest contig              | ERR2017411    | 516,770 | 394,993    | 278,191 |
| avg contig                  | ERR2017411    | 1,606.59 | 981.96    | 443.97 |
| n50                         | ERR2017411    | 4,063   | 2,820      | 1,620 |
| number of contigs           | ERR2017412    | 67,750  | 141,089    | 201,038 |
| largest contig              | ERR2017412    | 212,503 | 264,186    | 192,118 |
| avg contig                  | ERR2017412    | 1,360.63 | 807.96    | 340.48 |
| n50                         | ERR2017412    | 2,720   | 1,816      | 432   |

Table 3 Number of species predicted by each classifiers

| Classifier | Kaiju | CLARK | Kraken | MetaCache | MetaPhiAn2 | DUDes | GOTTCHA |
|------------|-------|-------|--------|-----------|------------|-------|---------|
|            | 26,666,674 paired-end reads (10s) length of 75bp |
| n/a        | 3553  | 372   | 346    | 50        | 10         | 9     | 10      |
| MEGAHIT    | 25    | 10    | 10     | 11        | 5          | 11    | 10      |
| MetaSPAdes | 31    | 10    | 10     | 15        | 3          | 13    | 10      |
| Ray        | 36    | 10    | 10     | 12        | 8          | 12    | 10      |
|            | 26,666,004 paired-end reads (100s) length of 75bp |
| n/a        | 3659  | 394   | 380    | 176       | 87         | 73    | 84      |
| MEGAHIT    | 1258  | 95    | 125    | 108       | 80         | 71    | 84      |
| MetaSPAdes | 1328  | 122   | 131    | 115       | 81         | 72    | 84      |
| Ray        | 2109  | 86    | 107    | 101       | 86         | 74    | 84      |
|            | 26,665,998 paired-end reads (400s) length of 75bp |
| n/a        | 3707  | 416   | 405    | 426       | 402        | 282   | 390     |
| MEGAHIT    | 2024  | 403   | 394    | 411       | 370        | 284   | 388     |
| MetaSPAdes | 2522  | 405   | 396    | 416       | 375        | 277   | 389     |
| Ray        | 754   | 398   | 392    | 394       | 10         | 188   | 99      |
|            | 17,853,919 paired-end reads (ERR2017411) length of 90bp |
| n/a        | 3654  | 3140  | 3638   | 1071      | 79         | 29    | 37      |
| MEGAHIT    | 2071  | 1477  | 1537   | 718       | 29         | 47    | 25      |
| MetaSPAdes | 2618  | 1782  | 1867   | 797       | 32         | 33    | 25      |
| Ray        | 2679  | 1630  | 1731   | 515       | 31         | 40    | 23      |
|            | 17,793,507 paired-end reads (ERR2017412) length of 90bp |
| n/a        | 3647  | 3075  | 3651   | 1044      | 82         | 48    | 45      |
| MEGAHIT    | 1653  | 1035  | 1058   | 611       | 23         | 33    | 26      |
| MetaSPAdes | 2312  | 1387  | 1423   | 679       | 39         | 42    | 29      |
| Ray        | 2192  | 1203  | 1297   | 448       | 21         | 21    | 22      |

Table 4 Pairwise similarity of a method to other methods

| Classifier | Kaiju | CLARK | Kraken | MetaCache | MetaPhiAn2 | DUDes | GOTTCHA |
|------------|-------|-------|--------|-----------|------------|-------|---------|
|            | 17,853,919 paired-end reads (ERR2017411) length of 90bp |
| n/a        | 0.66  | 0.69  | 0.66   | 0.68      | 0.65       | 0.82  | 0.80    |
| MEGAHIT    | 0.51  | 0.63  | 0.62   | 0.60      | 0.76       | 0.81  | 0.80    |
| MetaSPAdes | 0.53  | 0.65  | 0.64   | 0.63      | 0.73       | 0.81  | 0.81    |
| Ray        | 0.50  | 0.64  | 0.62   | 0.63      | 0.74       | 0.81  | 0.80    |
|            | 17,793,507 paired-end reads (ERR2017412) length of 90bp |
| n/a        | 0.66  | 0.69  | 0.65   | 0.68      | 0.71       | 0.82  | 0.80    |
| MEGAHIT    | 0.51  | 0.63  | 0.62   | 0.60      | 0.76       | 0.81  | 0.80    |
| MetaSPAdes | 0.53  | 0.65  | 0.64   | 0.63      | 0.73       | 0.81  | 0.81    |
| Ray        | 0.50  | 0.64  | 0.62   | 0.63      | 0.74       | 0.81  | 0.80    |