Sketching and sampling approaches for fast and accurate long read classification

Arun Das[1], Michael C. Schatz[1]

[1]Department of Computer Science, Johns Hopkins University, Baltimore, MD, 21218, USA

Contact: arun.das@jhu.edu, mschatz@cs.jhu.edu

Abstract

In modern sequencing experiments, identifying the sources of the reads is a crucial need. In metagenomics, where each read comes from one of potentially many members of a community, it can be important to identify the exact species the read is from. In other settings, it is important to distinguish which reads are from the targeted sample and which are from potential contaminants. In both cases, identification of the correct source of a read enables further investigation of relevant reads, while minimizing wasted work. This task is particularly challenging for long reads, which can have a substantial error rate that obscures the origins of each read.

Existing tools for the read classification problem are often alignment or index-based, but such methods can have large time and/or space overheads. In this work, we investigate the effectiveness of several sampling and sketching-based approaches for read classification. In these approaches, a chosen sampling or sketching algorithm is used to generate a reduced representation (a "screen") of potential source genomes for a query readset before reads are streamed in and compared against this screen. Using a query read's similarity to the elements of the screen, the methods predict the source of the read. Such an approach requires limited pre-processing, stores and works with only a subset of the input data, and is able to perform classification with a high degree of accuracy.

The sampling and sketching approaches investigated include uniform sampling, methods based on MinHash and its weighted and order variants, a minimizer-based technique, and a novel clustering-based sketching approach. We demonstrate the effectiveness of these techniques both in identifying the source microbial genomes for reads from a metagenomic long read sequencing experiment, and in distinguishing between long reads from organisms of interest and potential contaminant reads. We then compare these approaches to existing alignment, index and sketching-based tools for read classification, and demonstrate how such a method is a viable alternative for determining the source of query reads. Finally, we present a reference implementation of these approaches at https://github.com/arun96/sketching.

Keywords: Sketching, Sampling, Classification, MinHash, Metagenomics

1. Introduction

Metagenomics has become an increasingly popular area of study over the past two decades, and has enabled us to better understand the diversity, interactions and evolution of microbial communities in a plethora of environments [1–3]. Metagenomics has highlighted the problem of being able to quickly and accurately identify the source of a given DNA sequence from all the genomic material in a given sample. This is needed to classify and sort reads for further downstream analysis, and to identify and remove potential contaminants that are present in a sample.

The read classification problem is to identify the source genome of a given input read, usually by comparing the read to a list of potential source genomes and choosing the one with the highest similarity. This comparison may be done naively by comparing the entirety of each read to the entirety of each genome to find the best alignment or through an exhaustive analysis of k-mers present. While these approaches are highly accurate they can incur high computational overheads, which presents an opportunity for lower overhead techniques such as sketching or sampling, especially for long read data.

Sketching is the process of generating an approximate, compact summary of the data (a "sketch"), which retains properties of interest and can be used as a proxy for the original data [4]. Sampling selects a subset of the data, either systematically or randomly, but does not guarantee the preservation of these properties. Each has unique advantages: sketching has been shown to bound error better than sampling [5], while systematic sampling approaches (such as uniform sampling) can provide bounds on the number of samples from specific sections of the original data included in the generated subset. Both sketching and sampling provide simple routes to greatly reduce the size of an input set, while retaining the characteristics and features that identify the set, thus allowing a comparable level of accuracy.

One of the most well-known sketching approaches, and the main one we employ in our work, is MinHash, which was first presented as a method to estimate document similarity using the similarity between their hashed sub-parts [6]. It is now widely used in genomics, such as in Mash [7], which performs fast similarity and distance estimation between two input sequences, and tools such as Mash Screen [8] which uses MinHash to predict which organisms are contained in a mixture. Other tools include MashMap [9], which blends minimizers and MinHash for fast, approximate alignment of DNA sequences, and MHAP [10] to accelerate genome assembly. Beyond MinHash, several other related approaches have been proposed, such as bloom filters [11], the HyperLogLog sketch [12,13], and other sketching approaches to estimate similarity, containment or cardinality.

1.1. Approaches to Read Classification

The simplest approach to read classification is to simply align each query read to all potential source genomes, and use the genome with the best alignment as the predicted source. While the most accurate approach would be exhaustive sequence-to-sequence alignment with dynamic programming, this is impractically slow, so aligners typically use some form of seed-and-extend that start with exact matches and build out longer regions of high similarity. Two such aligners are Minimap2 [14] and Winnowmap [15], and these can be used to quickly generate accurate alignments over large sequences. However, alignment still remains computationally expensive, and offers a level of detail not always necessary in read classification.

A more sophisticated approach is index-based analysis, where a pre-computed index is created that aids in the classification of input reads against a chosen set of genomes. This index contains sequences that are specific to each genome or group of interest, and for each query read, the presence or absence of these pre-identified markers determines the classification of the read. The foremost examples of this form of read classification are the Kraken [16,17] set of tools, as well as tools such as CLARK [18] and Centrifuge [19]. While the read classification process in index-based approaches can be extremely fast, there is substantial time and space overhead associated with the construction of the index.

The space, time and computational overhead associated with alignment- and index-based read classification approaches has motivated the need for even faster, more accurate, and lower overhead alternatives. Sketching has proven to be an excellent answer to this problem, as the use of sketches instead

of whole genome comparisons provides the level of accuracy necessary for less exact tasks such as read classification, while substantially reducing overhead. Examples of this are MashMap [9] and MetaMaps [20], both of which use approximate similarity instead of exact alignment between regions of two sequences to perform alignment.

In this work, we present and critically analyze several methods that utilize sketching and sampling to reduce the computational overhead of read classification. We apply sketching, using MinHash- and minimizer-based approaches, as well as uniform sampling, to generate compact, approximate representations of potential source genomes for a given readset. We then classify reads against these representations, and demonstrate that we are able to classify, with a high degree of accuracy, reads from a microbial community and detect contaminants in real and simulated sequencing experiments.

2. Methods

In our methods, we utilize several sketching and sampling approaches to generate reduced representations of the source genomes, a "screen" of the genomes. Query reads are then compared against this screen, with the read being classified to the element most similar in the screen **(Figure 1)**.

2.1 Determining sketch and sample size

The biggest factor in such an approach is the size of the screen, meaning the fraction of the k-mers from the genomes that are recorded in the screen. The ideal screen size will minimize the input storage requirement while being large and detailed enough to capture the specificity of each genome. To do this, three main factors must be considered: (1) the size of the genomes; (2) the read length and error rate of the reads we are classifying; and (3) the amount of similarity needed to correctly match a read and its true source genome. We refer to this amount of similarity as the number "target matches" or "shared hashes", which is the number of sketched or sampled k-mers a read and its source genome share. In our work, we formalize this using the following formula:

$$Sketch/Sample\ Size = \frac{(\#\ Target\ Matches) \times (Genome\ Size)}{(Read\ Length) \times (1 - Read\ Error\ Rate)^{k}} \qquad \textbf{[EQ1]}$$

This formula allows us to sketch and sample at a rate where we expect to retain the target number of k-mers per read length of sequence in the original genome, adjusted for error. We adjust for error by computing the fraction of k-mers we expect to be affected by error at that error rate, and oversampling or oversketching to compensate for this.

The goal of sketching and sampling approaches is to reduce the space and computation overheads, while maintaining a comparable level of accuracy. These techniques achieve this goal by storing only a fraction of the k-mers from the input genomes, and by only comparing read k-mers to this selected set. As shown in Equation 1, the exact size of the screen depends on the experimental parameters. The compression factor is equal to the targeted number of k-mer matches per read divided by the read length, with an oversampling by a factor of $1/(1 - e)^k$ to correct for errors. This makes these approaches best suited for lower error, longer reads, as these require the fewest number of hashes from each read length, with shorter or higher error reads requiring larger screens. The number of target matches also determines screen size, but as we will see in the results section, the required number of target matches for an acceptable level of accuracy still creates a screen that is much smaller than the original data, especially for low error rate and longer reads. Since read k-mers are only compared against the stored k-mers in the screen, and never against the original sequence, these approaches greatly reduce the total number of comparisons that must be made in order to determine the source genome. These approaches also reduce the work involved in updating the set of potential source genomes; instead of rebuilding the whole index, sketches or samples can easily be added or removed from the screen.

2.2 Overview of sketching and sampling approaches

In this work, we consider several existing sketching and sampling techniques, along with a novel

clustering & comparison technique we describe below. A reference implementation of these approaches can be found at https://github.com/arun96/sketching.

Uniform. In this approach, k-mers are uniformly extracted across the genome to reach the desired sample size. The chief benefit of this approach is simplicity, including a guarantee on the distance between k-mers in our screen, which is generally not guaranteed for alternative approaches. This also guarantees that each read will have a highly predictable amount of overlap with the sampled version of its source genome, though error can obscure the detection of this overlap.

MinHash. This is a sketching technique used for quickly estimating the similarity between two input sequences by computing the Jaccard coefficient of the selected k-mers extracted from one sequence compared to those selected in a second sequence. There are several widely used methods to generate a MinHash sketch, such as using multiple hash functions or a partitioning of the space of possible k-mers. For our analysis, we use a single hash function, and select the n smallest hash values returned as is used in Mash and related works. This technique reduces the total amount of computation that must be done, and also allows us to just maintain a simple list of the lowest n values while discarding any higher values. As hashing is simply a permutation of the input values, this effectively generates a random sampling of n k-mers to be used as the representation of the original genome.

Weighted MinHash. The basic MinHash algorithm can be extended through the introduction of weights. The weight of each k-mer is a measure of the k-mer's "importance", with more highly weighted k-mers indicating a greater level of confidence in a match. Weights are typically based on the number of times that an element occurs, or on a predetermined scoring scheme. When comparing two sketches, these weights are used to generate the final similarity score; instead of just relying on the number of shared hashes between the sketches, we also consider the weight of the shared hashes. In our reference implementation, the weight is a measure of "uniqueness": a k-mer that occurs in just one of the potential source genomes, and therefore is a better identifier of the true source of a query sequence, is weighted more highly. As a baseline version, we compute the weight of a k-mer as the total number of genomes in our screen minus the number of genomes the k-mer is found in. Consequently, k-mers that are present in multiple source genomes receive lower weights, and k-mers that are unique to certain genomes, and therefore crucial in identifying the true source of a read are worth more. This scheme also helps break ties between very similar sequences: if a read shares the same number of k-mers with two potential source genomes, but shares a highly weighted k-mer with one of them, our approach will prefer the more likely source instead of relying on a random tie break. We further evaluate a "multiplier" into weighted MinHash, where k-mers that only occur in a single genome have their computed weight multiplied by some multiplier M (M=1 in regular weighted MinHash). This allows unique k-mers to play an even larger role in determining the similarity between a read and its genome.

Order MinHash. Just as the addition of weight can improve the accuracy of a MinHash based approach, consideration of the order of the retained minimal hashes can also help us filter out spurious matches and prioritize more likely sources for a query sequence. First presented as a method to improve estimation of the edit distance [21], an Order MinHash (OMH) sketch stores the selected n hashes in order sublists of L hashes, in the same order as they occur in the genome, with n/L lists making up the sketch. When two sketches are compared using Order MinHash, the algorithm checks which hashes are shared, along with if the shared hashes are in the correct order relative to each other. This method of comparing two sketches means that two sequences that contain the same k-mers but are rearranged versions of each other will have low similarity scores, while non-ordered MinHash would report high similarity. This approach is also more robust to sequencing errors than selecting a single long k-mer spanning the same distance.

Minimizer. Minimizers were originally proposed as a sequence compression method [22], but have become popular in genomics due to their ability to succinctly represent large sequences. In the most widely used form of a windowed-minimizer, the algorithm slides a window of size x over the sequence, and the k-mer with the smallest hash in that window is retained as the minimizer. This is repeated across the entire sequence, and the set of unique minimizers is used as the representation of the full sequence.

Similar to uniform sampling, using windowed-minimizers gives some guarantee on the distance between the retained k-mer hashes in our screen, as this distance is bounded above by twice the window size. Relatedly, for our minimizer-based approach, the window size is computed as the size of the genome divided by the desired number of k-mers per genome, multiplied by a fixed multiplier of two. The reasoning behind this multiplier is quite simple: since the distance between minimizers is uniformly distributed between 0 and the window size, we expect two minimizers per window. Consequently, to find n minimizers for a genome of size g, we simply double the window size. This effectively keeps the generated sketch and sample sizes relatively even across the approaches.

2.3 Clustering sketches

The approaches above create a screen of reduced representations for potential source genomes, where all input reads are compared against all elements of this screen. While this reduces the number of comparisons necessary to classify a read compared to traditional approaches, they still perform a large number of unnecessary comparisons with genomes with low similarity with the query read. To tackle this, we may want to adjust our approach to only perform in-depth comparisons against genomes that the read is likely to come from, and limit the number of comparisons with less-relevant genomes.

To do this, the algorithm first computes a hierarchical clustering on the input genomes, using a small sketch of each genome as its representation, and evaluating the similarity between these sketches. This clustering procedure groups together similar genomes, whose selected k-mers (and derivative reads) are more likely to be similar. The algorithm then uses the generated screen to populate the resulting clustering tree: each genome's reduced representation appears in the leaves of the tree, and the reduced representations are combined within internal nodes of the tree, until the root of the tree contains all the elements of the original screen. To limit the overhead of this approach to be comparable to those presented above, the algorithm downsamples the original elements of the screen as the algorithm constructs the tree. This downsampling can be done as a constant factor or by a factor proportional to the height of the tree, depending on the desired total size of the sketch tree.

Read classification is then performed by starting at the root, comparing the input read to the stored representations at each of the children of the root, and then descending into one or more children with sufficient representation. This process repeats until the algorithm arrives at a leaf, which is the genome predicted to be the source of the read. This approach quickly prunes genomes with low similarity to the input read, and focuses on the genomes that are likely to be the source of the read. These genomes are either from different but similar organisms, or different assemblies of the same genome.

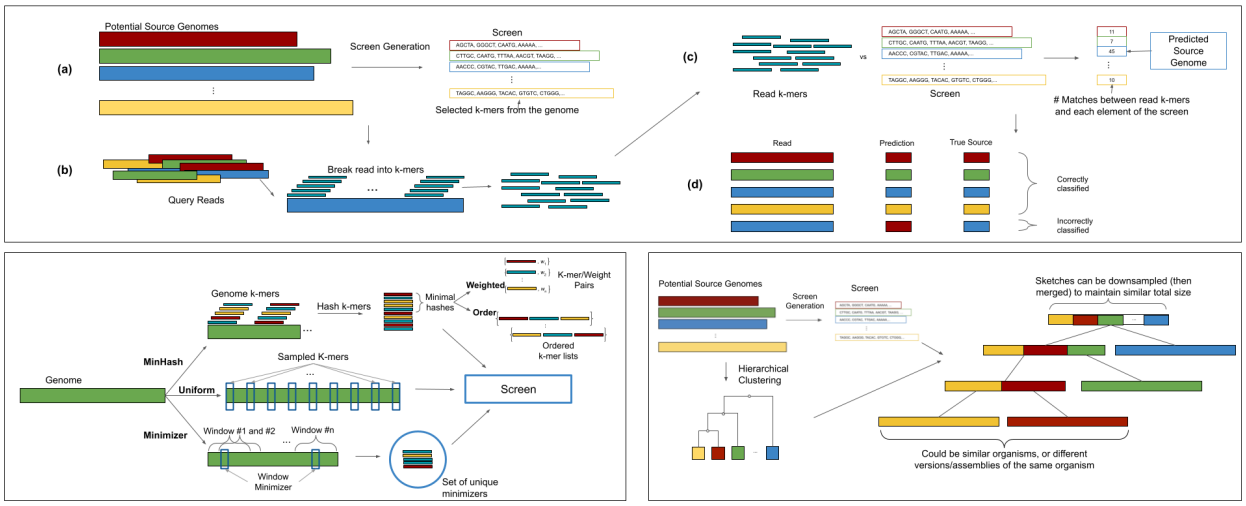


Figure 1. Overview of sketching and sampling methods. (Top) The screen is generated from potential input genomes, then read k-mers are extracted and compared against the screen, with the element of the screen most similar to the read being predicted as its source. **(Bottom Left)** The different sketching and

sampling approaches used to generate a screen. **(Bottom Right)** Sketch clustering approach: input genomes are clustered, and the generated screen is arranged to match this clustering, with reads compared to the root and then down the tree.

3. Results

3.1 Metagenomics Classification

Our main experimental results are based on the widely used Culturable Genome Reference (CGR) community of high quality microbial genomes sequenced from the human gut [23]. From this community, we selected all genomes that were available on RefSeq, giving us 1,310 genomes for our reference database, totalling 4.9 GBp of sequence. This community contains a range of genomes, including several clusters of highly similar genomes that make read classification more difficult (**Table 1**). This difficulty is especially true for approaches that work with reduced representations of the original genomes since unless the differences between these similar genomes are specifically captured, there will be no information available to distinguish between them.

As an example of a simpler community, we also analyze the ZymoBIOMICS Microbial Community Standards (ZYMO) and MBARC-26 [24] reference communities. When combined, these communities contain 34 distinct microbial and fungal genomes, totalling approximately 170Mbp of sequence. This community is much easier to classify within, as the genomes are relatively dissimilar (**Table 1**), and provides a baseline from which to interpret our results.

For our classification analysis, by default we use simulated PacBio HiFi-like reads that are 10Kb long with 1% error rates, with errors uniformly introduced. We simulate 10x coverage of each genome, yielding 4.8M reads for the CGR dataset, and 165K reads for the ZYMO+MBARC-26 dataset. For the classification, we use screen sizes that target 100 shared hashes with each of these reads, or on average one shared hash every 100bp based on EQ1. Results from a range of read lengths, error rates and screen sizes, as well as other experimental parameters, are reported later in this section, and presented in **Figure 2**. Reference implementations and analysis scripts, details on data availability, and scripts to benchmark existing tools can be found at https://github.com/arun96/sketching.

Community	Total Sequence	Number of Organisms	Number of Genomes with >X% Mash similarity to another genome in the community				
			X = 5%	X = 25%	X = 50%	X = 70%	X = 90%
Human Gut (CGR)	4.85GBp	1310	1218	1189	990	563	315
ZYMO + MBARC-26	170MBp	34	2	0	0	0	0

Table 1: Overview of the microbial communities, and similarity within them.

Classification Experiments

In microbial classification experiments reads are drawn from a microbial community, and compared against a screen generated from a reference database of known genomes. Under idealized conditions the database will contain reference genomes from all members of the community, although in practice the community may contain novel species or strains that are not yet characterized leading to poor matches or no matches at all. For simplicity, our reads are drawn from the reference database collection, and reads are then classified against the screen of all genomes. Accuracy is then measured as the fraction of reads correctly classified as being from the true source genome.

For the human gut microbe community, at our default experimental parameters, we see that all our sketching and sampling approaches achieve approximately 71-75% accuracy (**Figure 2**).We observe that around half of the genomes have classification accuracy over 90%, with the overall accuracy lowered by genomes that have high similarity with other members of the community. Specifically, we find that of the

genomes with less than 50% classification accuracy, 90% have another member of the community with which they are at least 70% similar (evaluated using Mash similarity). This implies that for these genomes, we expect 70% of their reads to be very similar to at least one other genome, greatly increasing the chances of each read being misclassified.

Just how disruptive highly similar genomes are to classification accuracy is visible when classifying reads from the simpler ZYMO+MBARC-26. Here, just two of the 34 members have Mash similarity >1.5% with each other, with those two members having a similarity of just 8% (**Table 1**). In our experiments, these two genomes provide the vast majority of misclassified reads, across read lengths and error rates. Overall, with a simpler community like this, any of these approaches achieve >99% classification accuracy, even with shorter reads and higher error rates.

Effect of experimental parameters on read classification

Read Length. We see increases in performance as read lengths get longer (**Figure 2a**), as we have more opportunities for the screen k-mers to match error-free k-mers in the read. Read length also affects the size of the screen, as longer reads mean smaller screens are necessary to achieve the desired number of shared hashes between a read and its source. Conversely, with shorter reads, the screen sizes must be larger to store the required number of k-mers to maintain the similar levels of accuracy.

Error Rates.We see decreases in accuracy at high error rates (**Figure 2a**), as fewer k-mers remain unaffected by error. This is accompanied by sharp increases in screen size, reducing the amount of compression we are able to achieve. With an error rate of 1% (as found in PacBio HiFi reads), we estimate that 81% of 21-mers will remain error free, while at an error rate of 5% (as is found in Oxford Nanopore reads) just 34% of the 21-mers will remain error free. This is even more pronounced at error rates close to 10% (as is found in CLR PacBio data and older Oxford Nanopore reads), where just 10% of the 21-mers can be expected to be unaffected by error. As our approach adjusts screen size to compensate for error rate, this results in extremely large screen sizes to compensate for high error rate.

Target number of shared hashes. The number of target matches determines how densely the reference genomes are represented, and therefore the size of the screen. Very low numbers of target matches result in reads being misclassified or unclassified, as there will not be enough similarity with the source genome. However, there is a plateau in performance as we increase the target number of shared hashes, as there are some sets of genomes that differ only in a small number of k-mers, and a sketching or sampling approach must draw from exactly those places in order to distinguish between them. We see steady increases in performance when increasing the target number of shared hashes up to 3 shared hashes every 200bp, but we see only minimal increases in performance beyond this (**Figure 2b**).

K. Based on testing a range of values for K (**Figure 2d**), we find that for k >20, while increasing k can result in minimal increases in performance, it also results in a larger increase in screen size. This is because longer k-mers have a higher chance of being affected by errors, so larger samples/sketches are necessary to ensure a robust number of error free k-mers remain. This was highly pronounced in our results, e.g. the step up from k=30 to k=50 came with a 1.3% increase in performance but a 22% larger screen. As we saw similar performance between $20 \leq k \leq 50$, we used k = 21 across our other experiments, as it provided the specificity necessary while keeping screen sizes small.

Weight. The addition of weight to traditional MinHash results in a slight increase in performance across read lengths and error rates. This is expected, as the discriminative k-mers now contribute more to the score and help break ties. Including a multiplier has a similar effect and more heavily weighting unique k-mers results in correctly breaking even more ties, resulting in another slight increase in performance. The addition of weight saw a 0.5% increase in classification accuracy, with the inclusion of a multiplier of 5 or 10 seeing a further 0.1% increase in performance (**Figure 2e**).

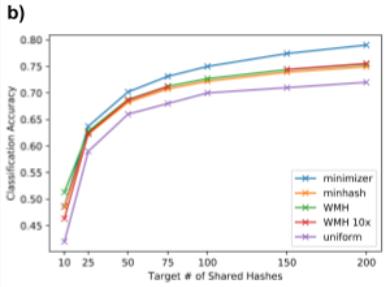
Order. Including order in a MinHash approach has a minimal impact on classification accuracy (**Figure 2e**). Order MinHash is initially proposed as a metric for estimating edit distance, and therefore is most beneficial when determining the similarity between rearranged strings that cannot be distinguished by an unordered MinHash. With read classification from this large set of microbial genomes, such rearrangements are not common, and thus order helps in only a few cases.

Cluster Downsampling Rate. Using MinHash screens, we compared the accuracy across three approaches to clustering: (1) screens downsampled by a constant factor; (2) screens downsampled based on their height in the sketch tree; and (3) screens that are not downsampled at all. We find that constant factor downsampling approaches, with factors 2 and 4, maintain a good degree of accuracy (71% and 70% respectively, compared to the 72.8% accuracy with MinHash), while keeping the number of comparisons similar to or less than the original MinHash approach. Height-based downsampling approach results in a sharp drop in accuracy (62%), as the screens near the root of the tree are downsampled to the point where discriminative k-mers are lost.

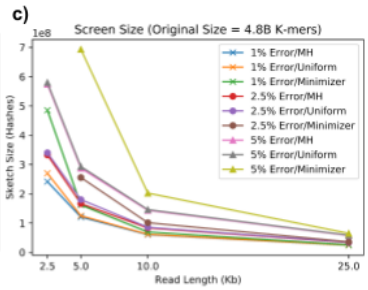
a)

		Read Lengths (Kb)			
		2.5	5.0	10.0	25.0
MinHash	Read Error (%) 0.0	65.0	69.0	72.8	77.5
	1.0	64.6	68.9	72.8	77.6
	2.5	64.1	68.6	72.7	77.8
	5.0	62.7	67.8	72.4	77.7
	10.0	62.0	66.9	70.8	77.2
Minimizer	Read Error (%) 0.0	70.0	74.0	77.4	82.1
	1.0	69.2	72.7	75.6	79.8
	2.5	69.0	72.1	74.2	78.0
	5.0	68.9	72.3	74.0	76.7
	10.0	67.4	71.9	73.6	78.6
Uniform	Read Error (%) 0.0	63.9	68.1	71.1	74.7
	1.0	63.5	67.6	70.9	73.7
	2.5	63.1	67.1	70.5	72.9
	5.0	62.4	66.4	69.5	72.6
	10.0	61.9	66.0	68.9	72.1

b)



c)



d)

K	10	15	20	30	35	40	50
Accuracy (%)	28.5	48.0	72.6	72.9	73.3	73.6	73.9
Sketch Size (10M Hashes)	5.4	5.7	6.0	6.6	7.0	7.3	8.1

e)

	MH	WMH w/ Multiplier					OMH
		1	2	5	10	15	
Accuracy (%)	72.2	72.7	72.7	72.8	72.8	72.8	72.4

Figure 2. Key results across parameters on the gut microbial dataset. (a) Results across a range of read lengths and error rates, **(b)** the effect on accuracy of increasing the target number of shared hashes, **(c)** The impact read length and error have on sketch size across our approaches, **(d)** The performance of a MinHash approach across a range of k values, and **(e)** the impact of incorporating weights, a range of weight multipliers and order into MinHash.

3.2 Host Contaminant Detection

The goal of the contaminant detection and classification is to, for a given read set, distinguish between reads that come from organisms or sequences of interest, and reads that are from potential contaminants. For our experiments, the organism of interest was a human, with human reads being drawn from GRCh38 [25] and mixed with contaminant reads drawn from a selected microbial community, and classified against a screen containing both the human and microbial genomes. The sequence reads were simulated as above using 10kbp reads at 10x coverage with 1% error. Accuracy is measured as the fraction of human and microbial reads correctly identified as being of interest or as being a contaminant, while also measuring the fraction of contaminant reads that are correctly classified to their source genome.

When using the human gut microbe community as the source of contaminant reads, we find all the sketching and sampling approaches to be successful at distinguishing between microbial and human reads. Across all approaches, more than 99% of all human reads are correctly distinguished from microbial reads and classified to the chromosome they are drawn from. We also observe that very few microbial reads are misclassified as human, with over 99% correctly identified as being contaminants. This is not unexpected; human and microbial genomes are quite dissimilar, and therefore the sketches or

samples of the sequences will also be dissimilar, making read classification successful in nearly all cases. However, it is worth highlighting that we are able to distinguish between these sequences while storing just 2% of the original k-mers.After distinguishing between reads and contaminants, we then attempt to classify the reads from the contaminant microbes to the exact source genome. The results match what we present in the classification experiments; of the microbial reads that are recognized as contaminants, approximately 75% are mapped to the exact genome, with the misclassified reads coming from the genomes discussed in the previous section. We also classify human reads to the chromosome they are drawn from, and are able to do this with >95% accuracy.

3.3 Comparison to existing tools

To evaluate the performance of the sketching and sampling approaches, we also tested several widely used approaches for read classification on the same dataset and experimental settings. As in the previous experiments, accuracy in classification experiments is measured as the number of reads correctly mapped back to the microbial genomes they were drawn from, and accuracy in contaminant detection experiments is measured as the number of human and microbial reads identified.

Alignment-based

To test the effectiveness of alignment-based approaches to read classification, we test two widely used aligners, Minimap2 [14] and Winnowmap [26]. Minimap2 uses query minimizers as seeds for the alignment, while Winnowmap2 adds a preprocessing step to downweight commonly occurring minimizers to reduce the chance of them being selected. In both approaches, we align our read sets against the genomes of the selected community, and calculate the predicted source of the read as the sequence to which it is mapped. For microbial classification, we find that both these tools perform slightly better than our MinHash and minimizer based approaches. Compared to an accuracy of 77% and 79% in our MinHash and minimizer approaches with two shared hashes every 100bp, Minimap2 and WinnowMap both achieve an accuracy of 81% (**Table 2**). Both alignment approaches achieve low accuracy on the same genomes that our sketching and sampling approaches struggle on; namely, genomes with high-similarity relatives in the community. For contaminant detection, both alignment-based tools are able to correctly distinguish 99.5% of the human and contaminant reads.

Index-based

Kraken2 [17] utilizes a preprocessed index of shared k-mers to determine the source of a query sequence. Each k-mer in the query sequence is classified to an element in the index, and we determine the source of the whole sequence as the element in the database to which a plurality of k-mers are assigned. When using both a pre-built RefSeq database and a custom database built over our test community, we find that genome level identification is difficult between the highly similar members of our community (**Table 2**). We observe large numbers of misclassifications between reads from these similar genomes, as well as Kraken2 classifying some of these reads only to common ancestors, and not to one of the genomes. For genomes without similar members in the community, the majority of their reads are correctly classified to a single genome, giving Kraken2 an overall classification accuracy of 72%. When it comes to distinguishing between human and microbial reads, Kraken2 is able to correctly identify >99% of the reads. This is in line with the sampling and sketching results, as the k-mer compositions of human and microbial genomes are quite different.

Centrifuge [19] uses an index built on the compressed genomes of its target sequences to classify query sequences. Centrifuge's read classification is similar to Kraken2 and with genomes that have no similar members of the community, it is able to accurately classify reads as coming from a single genome. However, with genomes that have highly similar counterparts in the community, Centrifuge has high rates of misclassification, or the algorithm will not classify to a single genome, instead leaving the read unclassified or classifying to a common ancestor. This results in Centrifuge classifying 72% of microbial reads classified back to the correct genome. Contamination detection is accurate with Centrifuge and more than 99.5% of human reads and microbial reads are identified.

CLARK [18] uses a pre-compiled list of discriminative k-mers for the community it is indexing, and performs classification of query sequences based on similarity to this list. While there are still misclassifications and unclassified reads at rates comparable to other tools, CLARK's use of discriminative k-mers slightly reduces the impact of highly similar genomes in the community, allowing it to identify the few differences between them, and helping it to achieve a classification accuracy of 73.5%. For contaminant detection, CLARK is also able to distinguish 99.5% of both human and microbial reads.

Sketching-based

Finally, we test the effectiveness of an existing sketching-based approach to read classification by benchmarking MashMap [9]. MashMap computes alignments by estimating k-mer based Jaccard similarity between query sequences with MinHash sketches. Query reads are aligned against a list of potential source genomes, and accuracy is computed as the fraction of reads that are correctly assigned back to their source genome.

We find that MashMap performs worse than classic alignment-based approaches, and similarly to our MinHash approaches, with 74.5% classification accuracy on the gut microbe dataset. Alignment boundaries in MashMap are determined through the Jaccard similarities of sketches. As a result, just as in the MinHash approach, it is susceptible to misclassifications between highly similar genomes, which have high Jaccard similarities with the same reads. For contaminant detection**,** MashMap is able to distinguish >99% of the human and microbial reads. This is not unexpected, as human and microbial sketches are expected to be largely dissimilar, making Jaccard similarity-based assignment easier.

	Microbial Classification	**Contaminant Detection**	
	Accuracy (%)	**% of Human reads identified**	**% of microbial reads identified**
Minimap2	81.3	99.5	99.5
WinnowMap	81.3	99.5	99.5
Kraken2 *(RefSeq DB)*	72.0	99.3	99.2
Kraken2 *(Custom DB)*	72.2	99.3	99.3
Centrifuge *(RefSeq DB)*	72.2	99.3	99.2
Centrifuge *(Custom DB)*	72.4	99.3	99.3
CLARK	73.5	99.5	99.5
MashMap	74.5	99.5	99.4

Table 2: Performance of existing tools. We find that alignment-based approaches are better at genome-level read classification, with the index- and sketching-based approaches less able to distinguish between highly similar genomes. All tools perform similarly in contaminant detection.

3.4 Analysis of genuine metagenomics sequencing data

To test the accuracy of our approaches on real sequencing data, we mapped PacBio HiFi reads from the Human Gut Microbiome Pooled Standards [27] to the CGR community database. For this analysis, we used 100K reads with length averaging ~10Kb and median quality of ~Q40. We first aligned these reads using Minimap2, and found Minimap2 aligned 78,137 reads to the CGR community, with 50,640 reads having an alignment length >5Kb. Without this alignment threshold, almost 26,000 more reads are mapped to multiple genomes. For these reads, we take the sequence with the longest alignment as the source of the read.

We then classify these reads using our sketching and sampling approaches against a generated screen of the CGR community, built for 10KB, 1% error reads and 100 shared matches per read. We consider a read to be mapped if it has at least 5 shared hashes with the genome it is classified to. With this threshold, approximately 60,000 reads are classified in each of the sketching and sampling approaches, with approximately 25% of the classified reads tied between multiple sources (**Table 3**).

We benchmark these classification results against the alignments generated with Minimap2. Our classification results agree with approximately 60% of the reads classified by Minimap2 with no minimum alignment length, and approximately 80% of the reads classified with a minimum alignment length of 5Kb (**Table 3**). This 20% increase in consistency between the classification calls is expected, since adding a minimum alignment length limits the Minimap2 classification to reads that are aligned with more certainty, and these are reads that are more likely to share a significant amount of similarity with the generated screen. Without this threshold, reads with minimal similarity to a source genome can be classified based just on small regions of alignment. These reads are unlikely to share a significant number of hashes with any elements in the screen, resulting in the sketching and sampling approaches not classifying them, or having to randomly break ties between multiple low scoring genomes.

	Total number of reads classified	**Number of reads classified to multiple genomes**	**Number of reads with same prediction as Minimap2**	
			No Threshold	**>5Kb Alignments**
Minimap2 (No alignment threshold)	78,137	25,955	N/A	
Minimap2 (>5Kb alignments)	50,640	1,228	N/A	
MinHash	60,724	15,911	46,506	41,150
Minimizer	59,578	16,616	47,105	41,455
Uniform	63,155	16,310	42,396	39,873

Table 3: Performance on real sequencing data. Comparison of the classification of our sketching and sampling approaches against Minimap2 classifications, with and without a minimum alignment length.

4. Discussion

Existing approaches for read classification often incur high computational overheads. In this work, we presented and analyzed a range of sketching and sampling approaches for read classification, designed to minimize these overheads. The techniques presented here are able to achieve comparable accuracy to existing read classification methods, with all approaches correctly distinguishing reads from dissimilar genomes but struggling with the classification of reads from highly similar genomes. Alignment-based approaches are slightly better suited to handling these, as they do direct comparisons of the reads against the source genomes, with k-mer indexes and sketching-based methods struggling to narrow down the exact source between several similar sequences.

Sketching and sampling approaches are able to perform read classification with minimal preprocessing, and while storing only a subset of the data, indicating that not all the information traditional read classification utilizes is necessary, and that there will still be edge cases that even preprocessing the dataset or having access to the full data cannot solve. These methods are best suited for longer, low-error reads, and incur a higher footprint and decreased performance when classifying shorter, higher error rate reads. Future work could be into sketching and sampling techniques better suited for high error rate environments, such as the use of gap k-mers [28] to increase error tolerance, as well as research into more informed techniques.

5. Acknowledgements

We would like to thank Benjamin Langmead, Daniel Baker and Bohan Ni for their help. This work was supported in part by National Science Foundation (NSF) grants DBI-1627442, IOS-1732253, and IOS-1758800, National Institutes of Health (NIH) grant U01CA253481, and the Mark Foundation for Cancer Research (19-033-ASP) to M.C.S. This work utilized the computational resources of the Maryland Advanced Research Computing Center (https://www.marcc.jhu.edu/).

References

1. Quince C, Walker AW, Simpson JT, Loman NJ, Segata N. Shotgun metagenomics, from sampling to analysis. Nat Biotechnol. 2017;35: 833–844. doi:10.1038/nbt.3935

2. Handelsman Jo. Metagenomics: Application of Genomics to Uncultured Microorganisms. Microbiol Mol Biol Rev. 2004;68: 669–685. doi:10.1128/MMBR.68.4.669-685.2004

3. Breitwieser FP, Lu J, Salzberg SL. A review of methods and databases for metagenomic classification and assembly. Brief Bioinform. 2019;20: 1125–1136. doi:10.1093/bib/bbx120

4. Rowe WPM. When the levee breaks: a practical guide to sketching algorithms for processing the flood of genomic data. Genome Biol. 2019;20: 199. doi:10.1186/s13059-019-1809-x

5. Cormode G. Data sketching. Commun ACM. 2017;60: 48–55. doi:10.1145/3080008

6. Broder AZ. On the resemblance and containment of documents. Proceedings Compression and Complexity of SEQUENCES 1997 (Cat No 97TB100171). IEEE; 1997. pp. 21–29. Available: https://ieeexplore.ieee.org/abstract/document/666900/

7. Ondov BD, Treangen TJ, Melsted P, Mallonee AB, Bergman NH, Koren S, et al. Mash: fast genome and metagenome distance estimation using MinHash. Genome Biol. 2016;17: 132. doi:10.1186/s13059-016-0997-x

8. Ondov BD, Starrett GJ, Sappington A, Kostic A, Koren S, Buck CB, et al. Mash Screen: high-throughput sequence containment estimation for genome discovery. Genome Biol. 2019;20: 232. doi:10.1186/s13059-019-1841-x

9. Jain C, Dilthey A, Koren S, Aluru S, Phillippy AM. A Fast Approximate Algorithm for Mapping Long Reads to Large Reference Databases. J Comput Biol. 2018;25: 766–779. doi:10.1089/cmb.2018.0036

10. Berlin K, Koren S, Chin C-S, Drake JP, Landolin JM, Phillippy AM. Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. Nat Biotechnol. 2015;33: 623–630. doi:10.1038/nbt.3238

11. Solomon B, Kingsford C. Fast search of thousands of short-read sequencing experiments. Nat Biotechnol. 2016;34: 300–302. doi:10.1038/nbt.3442

12. Baker DN, Langmead B. Dashing: fast and accurate genomic distances with HyperLogLog. Genome Biol. 2019;20: 265. doi:10.1186/s13059-019-1875-0

13. Breitwieser FP, Baker DN, Salzberg SL. KrakenUniq: confident and fast metagenomics classification using unique k-mer counts. Genome Biol. 2018;19: 198. doi:10.1186/s13059-018-1568-0

14. Li H. Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics. 2018;34: 3094–3100. doi:10.1093/bioinformatics/bty191

15. Jain C, Rhie A, Hansen N, Koren S, Phillippy AM. A long read mapping method for highly repetitive reference sequences. bioRxiv. 2020. p. 2020.11.01.363887. doi:10.1101/2020.11.01.363887

16. Wood DE, Salzberg SL. Kraken: ultrafast metagenomic sequence classification using exact

alignments. Genome Biol. 2014;15: R46. doi:10.1186/gb-2014-15-3-r46

17. Wood DE, Lu J, Langmead B. Improved metagenomic analysis with Kraken 2. Genome Biol. 2019;20: 257. doi:10.1186/s13059-019-1891-0

18. Ounit R, Wanamaker S, Close TJ, Lonardi S. CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. BMC Genomics. 2015;16: 236. doi:10.1186/s12864-015-1419-2

19. Kim D, Song L, Breitwieser FP, Salzberg SL. Centrifuge: rapid and sensitive classification of metagenomic sequences. Genome Res. 2016;26: 1721–1729. doi:10.1101/gr.210641.116

20. Dilthey AT, Jain C, Koren S, Phillippy AM. Strain-level metagenomic assignment and compositional estimation for long reads with MetaMaps. Nat Commun. 2019;10: 3066. doi:10.1038/s41467-019-10934-2

21. Marçais G, DeBlasio D, Pandey P, Kingsford C. Locality-sensitive hashing for the edit distance. Bioinformatics. 2019;35: i127–i135. doi:10.1093/bioinformatics/btz354

22. Roberts M, Hayes W, Hunt BR, Mount SM, Yorke JA. Reducing storage requirements for biological sequence comparison. Bioinformatics. 2004;20: 3363–3369. doi:10.1093/bioinformatics/bth408

23. Zou Y, Xue W, Luo G, Deng Z, Qin P, Guo R, et al. 1,520 reference genomes from cultivated human gut bacteria enable functional microbiome analyses. Nat Biotechnol. 2019;37: 179–185. doi:10.1038/s41587-018-0008-8

24. Singer E, Andreopoulos B, Bowers RM, Lee J, Deshpande S, Chiniquy J, et al. Next generation sequencing data of a defined microbial mock community. Scientific Data. 2016. doi:10.1038/sdata.2016.81

25. Schneider VA, Graves-Lindsay T, Howe K, Bouk N, Chen H-C, Kitts PA, et al. Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. Genome Res. 2017;27: 849–864. doi:10.1101/gr.213611.116

26. Jain C, Rhie A, Zhang H, Chu C, Walenz BP, Koren S, et al. Weighted minimizer sampling improves long read mapping. Bioinformatics. 2020;36: i111–i118. doi:10.1093/bioinformatics/btaa435

27. Data release: Human microbiome samples demonstrate advances in HiFi-enabled metagenomic sequencing. 2 Aug 2021 [cited 5 Nov 2021]. Available: https://www.pacb.com/blog/data-release-human-microbiome-samples-demonstrate-advances-in-hifi-enabled-metagenomic-sequencing/

28. Ghandi M, Mohammad-Noori M, Beer MA. Robust k-mer frequency estimation using gapped k-mers. J Math Biol. 2014;69: 469–500. doi:10.1007/s00285-013-0705-3