Sequence Analysis

KMCP: accurate metagenomic profiling of both prokaryotic and viral populations by pseudo-mapping

Wei Shen[1,*], Hongyan Xiang[1], Tianquan Huang[1], Hui Tang[1], Mingli Peng[1], Dachuan Cai[1], Peng Hu[1,*], Hong Ren[1,*]

[1]Key Laboratory of Molecular Biology for Infectious Diseases (Ministry of Education), Institute for Viral Hepatitis, Department of Infectious Diseases, The Second Affiliated Hospital, Chongqing Medical University, Chongqing 400010, China

*To whom correspondence should be addressed.

Abstract

Motivation: The growing number of microbial reference genomes enables the improvement of metagenomic profiling accuracy but also imposes greater requirements on the indexing efficiency, database size, and runtime of taxonomic profilers. Additionally, most profilers focus mainly on bacterial, archaeal, and fungal populations, while less attention is paid to viral communities.

Results: We present KMCP, a novel k-mer-based metagenomic profiling tool that introduces genomic positions to nucleotide k-mers by splitting the reference genomes into chunks and then stores k-mers in a modified and optimized COBS index for fast alignment-free sequence searching. The index size of KMCP is smaller than that of COBS, and the batch searching speed is increased by 10 times. KMCP combines k-mer similarity and genome coverage information to reduce the false positive rate of k-mer-based taxonomic classification and profiling methods. Benchmarking results based on simulated and real data demonstrate that KMCP not only allows the accurate taxonomic profiling of prokaryotic and viral populations but also provides confident pathogen detection in clinical samples of low coverage.

Availability and Implementation: The software is open-source under the MIT license and available at https://github.com/shenwei356/kmcp.

Contact: shenwei356@cqmu.edu.cn (W., Shen); hp_cq@163.com, hupengcq@hospital.cqmu.edu.cn (P., Hu); renhong0531@cqmu.edu.cn (H., Ren)

Keywords: Metagenomic profiling, K-mer, Genome chunks, Pseudo-mapping

1 Introduction

High-throughput sequencing technologies and computational methods have been combined in shotgun metagenomics approaches to study the composition and dynamics of microbial communities composed of bacteria, archaea, viruses, and fungi (Quince *et al.*, 2017). Metagenomics is an essential means of exploring microbial diversity (Caporaso *et al.*, 2012), the

interaction between microorganisms and hosts (Human Microbiome Project, 2012), and the rapid detection of uncultivated and difficult-to-cultivate pathogens (Chiu and Miller, 2019). Metagenomic sequencing is mainly based on next-generation high-throughput sequencing platforms, which can generate millions of 50-250 bp short reads from each sample. Reference-based metagenomic profiling tools map these sequencing reads to a reference database and resolve the species present in the sample and their relative abundances (Sczyrba *et al.*, 2017).

Several studies have assessed metagenomic profiling tools for taxonomic classification and profiling (Meyer *et al.*, 2022; Sczyrba *et al.*, 2017; Sun *et al.*, 2021; Ye *et al.*, 2019). Based on the methodology involved, these tools can be categorized into three types: (1) k-mer (length-k subsequences) based, such as Kraken2 (Wood *et al.*, 2019), Bracken (Lu *et al.*, 2017), Centrifuge (Kim *et al.*, 2016) and Ganon (Piro *et al.*, 2020); (2) alignment-based, such as DIAMOND (Buchfink *et al.*, 2015), Kaiju (Menzel *et al.*, 2016), DUDes (Piro *et al.*, 2016), and SLIMM (Dadi *et al.*, 2017); and (3) marker-gene based, such as MetaPhlAn2 (Truong *et al.*, 2015) and mOTUs2 (Milanese *et al.*, 2019). K-mer-based methods utilize exact k-mer matching, which is fast but sensitive to genomic mutation and variation because a single point mutation will result in at most *k* mismatches of k-mers. The taxonomic assignment algorithm based on the lowest common ancestor (LCA) often shows low accuracy at low taxonomic ranks because of closely related species sharing a large fraction of k-mers, and this accuracy decreases further with the increasing number of reference genomes (Nasko *et al.*, 2018). In contrast, alignment-based methods show high tolerance for base variation. Marker gene approaches perform well in the identification of archaea and bacteria (Meyer *et al.*, 2022), while it is difficult to accurately identify viruses with this strategy since viruses do not have universally conserved genes, such as the 16S and 18S rRNA genes (Breitwieser *et al.*, 2019).

The accuracy of metagenomic profiling also depends on the size and quality of the reference databases, and more extensive and comprehensive databases tend to yield better results (Breitwieser *et al.*, 2019; Nasko *et al.*, 2018; Piro *et al.*, 2020). Databases such as NCBI RefSeq (O'Leary *et al.*, 2016) and GenBank (Sayers *et al.*, 2022) contain increasing numbers of microbial genomes (e.g., 250,000 assemblies belonging to 48,000 species in RefSeq release 210), which poses a major challenge to taxonomic profilers in terms of database size and construction efficiency. Recently, several genome catalogs have been created with the aim of providing comprehensive, high-quality reference genomes for taxonomic and functional research, such as human gut prokaryotic genome collections (UHGG (Almeida *et al.*, 2021), HumGut (Hiseni *et al.*, 2021), and an archaeal catalog (Chibani *et al.*, 2022)) and human gut virus catalogs (GVD (Gregory *et al.*, 2020), GPD (Camarillo-Guerrero *et al.*, 2021), and MGV (Nayfach *et al.*, 2021)). Sourmash (Irber *et al.*, 2022) outperformed other tools in the Critical Assessment of Metagenome Interpretation (CAMI) based on mouse gut datasets (Meyer *et al.*, 2019) using FracMinHash sketches with a scale of 10,000 calculated from all 141,677 prokaryotic genomes in a RefSeq snapshot. The sketching algorithm uses a small number of signatures from whole-genome sequences; therefore, it can index a large number of reference genomes. However, it is limited by reduced sensitivity in small genomes such as those of viruses.

Several computational approaches have been introduced to index and query large collections of sequencing datasets and genomes over the last few years (Marchet *et al.*, 2021). K-mer aggregative methods index each dataset using a separate Bloom filter and then build an aggregation data structure for querying. Sequence Bloom Tree (SBT) approaches (Harris and

Medvedev, 2020; Solomon and Kingsford, 2016; Solomon and Kingsford, 2018; Sun *et al.*, 2018) are designed for collections with high k-mer redundancy, such as human RNA-seq sequencing data. In contrast, Bloom filter matrix-based methods, including the BItsliced Genomic Signature Index (BIGSI) (Bradley *et al.*, 2019), Compact Bit-Sliced Signature Index (COBS) (Bingmann *et al.*, 2019), and Interleaved Bloom filter (IBF) (Dadi *et al.*, 2018) approaches, are suitable for heterogeneous k-mer sets, such as those of microbial genomes. Instead of performing inexact pattern matching in alignment methods, k-mer-based methods simply examine the fraction of matched k-mers in the query sequence. Although the Bloom filter may return false positives due to its probabilistic data structure, the false-positive rate of a query sequence decreases exponentially with the number of query k-mers (Bingmann *et al.*, 2019; Solomon and Kingsford, 2016). These features indicate that k-mer-based Bloom filter matrix methods can be applied to index reference genomes for taxonomic profiling. Among these methods, IBF has been used for read mapping in DREAM-Yara (Dadi *et al.*, 2018) and for read classification in Ganon (Piro *et al.*, 2020).

Classical k-mer-based taxonomic classification methods (e.g., Kraken2) map each k-mer to the taxonomic identifier (TaxId) and assign reads to the LCA. However, assigning an ambiguous read to the LCA could result in a low classification resolution, usually above the species rank. The SLIMM (Dadi *et al.*, 2017) alignment method performs read mapping with Bowtie2 (Langmead and Salzberg, 2012) and maps matched reads into bins of fixed width across the reference genome; then, it utilizes the coverage information of reference genomes to remove unlikely genomes and obtain more uniquely mapped reads for assignment at lower ranks.

The increasing number of reference genomes calls for new taxonomic profiling tools with high efficiency in database building and high accuracy in prokaryotic and viral populations. Here, we present a novel taxonomic profiling tool, KMCP (K-mer-based Metagenomic Classification and Profiling), that uses a modified COBS data structure to index reference genomes and combines k-mer similarity and genome coverage information to improve taxonomic profiling accuracy. We evaluate KMCP based on simulated and real metagenomic datasets composed of data from archaea, bacteria, and viruses, and the benchmarking results indicate that KMCP performs well in both prokaryotic and viral populations.

2 Methods

2.1 Indexing

KMCP efficiently builds a database from a collection of genome sequences and taxonomic information. The microbial genomes are split into ten (for archaea, bacteria, and fungi) or five (for viruses) chunks with 100-bp overlaps, and the k-mer location information is further utilized in taxonomic profiling. For genomes without a single complete genome sequence, chromosomes or contigs are concatenated with intervals of k-1 bases of N to avoid introducing fake k-mers. For each genome, nonredundant nucleotide canonical k-mers (k=21, by default) are hashed into 64-bit unsigned integers with ntHash (Mohamadi *et al.*, 2016). Then, k-mer hashes alongside the genome chunk number and genome size are saved into binary files with the extension *.unik*. Next, these k-mer hashes are stored in a modified COBS index. Briefly, all *.unik* files are sorted by the number of k-mer hashes and divided into blocks of N genome chunks. Then, N Bloom filters of equal size are constructed and concatenated in a columnwise manner into a byte matrix for each block. Based

on a given number of hash functions and a desired false-positive rate, the size of the Bloom filters is determined by the number of k-mers in the largest genome chunk. To avoid generating extremely large Bloom filters for the last block with the largest genome chunks, we set three thresholds to create extra small indexes with fewer chunks, resulting in a smaller database size than COBS with the same parameters. In addition, the index files of all blocks are not concatenated to a large one for the parallelization of searching. These index files and taxonomic information, including NCBI taxonomy dump files and TaxId mapping files, comprise the KMCP database.

2.2 Searching

For sequence searches against the KMCP database, k-mers are computed with the parameters of the database and sent to query against Bloom filters of all index files in parallel. Similar to COBS, index files are mapped onto virtual address space using *mmap* by default or loaded into memory. Although COBS shows a good query performance relative to similar tools, it is insufficient for shotgun metagenomic sequencing analysis. In our implementation, steps including index file loading, k-mer computation and hashing, and Bloom filter querying are highly parallelized to take the full advantage of modern multiple-core CPUs and high-speed solid-state disks.

Many optimizations are performed to accelerate batch querying. Multiple hashing functions are optionally used to reduce the false-positive rate of Bloom filters, which requires a bitwise AND step to double-check the existence of a k-mer. Instead of the byte-by-byte operation performed in COBS, the AND step is optimized with Single-Instruction Multiple-Data (SIMD) instructions. For the ADD step, which cumulatively adds the number of matched k-mers in a Bloom filter, COBS uses array lookup to avoid unpacking the bit-packed count information. However, array lookup requires frequent memory access, which is fast but not free. Here, we use a high-performance vectorized positional *popcount* method (https://github.com/clausecker/pospop), which utilizes SIMD instructions to unpack and count k-mers in CPU registers for a list of bytes. To balance the speed of memory access for collecting columnwise bytes and invoking the *pospop* function, we set the size of the byte list to 64 (i.e., counting matched elements for 64 k-mers in 8 (8 bits for a byte) genome chunks in one function invocation). In addition, sixty-four is the cache line size for most 64-bit computers, which helps to reduce cache misses and provides extra speedup. For the left k-mers (<64) of the query sequence, we unroll the loops of 63 possible k-mer sizes to increase CPU instruction execution efficiency. Furthermore, many language-level optimizations are performed to reduce memory occupation and accelerate searching speed.

Due to mutation or intraspecific polymorphism, a low k-mer similarity threshold is needed to identify querying sequences from distant homologous genomes during the search. At the same time, the Bloom-filter-based probabilistic index required enough matched k-mers to reduce the false-positive rate. To balance the index size, searching accuracy, and searching speed, we set the desired false-positive rate of the Bloom filters as 0.3 and use one hash function by default, following BIGSI and COBS. During searching, the threshold of k-mer coverage (proportion of matched k-mers and unique k-mers of a query) is set to 0.55 (approximately equal to a sequence identity of 96.5%, Supplementary Fig. S1d), which limits the false-positive probability of a query to a reasonable level according to formula (5) in (Solomon and Kingsford, 2016) (e.g., 0.003 for 150-bp reads with 83 k-mers matched and 0.028 for 100-bp reads with 55 k-mers matched).

2.3 Scalable searching

By default, KMCP spawns one searching process for each index file. When sufficient CPU cores are available, KMCP automatically generates more processes to exploit the CPU resources. In addition, KMCP assigns a unique ascending number to each query read and outputs them into the search result, which can be used to merge search results against multiple databases. Since the search speed is linearly related to the number of reference genome chunks in the databases and the index data (Bloom filters) of all genome chunks are independent, we can build smaller databases from parts of the reference genomes and merge the results after searching against all small databases. The searching step can be parallelized with a computer cluster in which each computation node searches against a small database, and the additional merging step is rapid. Computers with limited main memory can also utilize an extensive collection of reference genomes by building and searching against small databases.

2.4 Profiling

In the indexing phase, reference genomes are split into 10 or 5 chunks. The coverage information of all chunks help determine the existence of a genome, thus reducing false positives (Dadi *et al.*, 2017). To implement this strategy, a three-round profiling method is applied (Supplementary Fig. S2). For the first round, reference genomes with at least one uniquely matched read are collected and filtered by the genome chunk fraction; specifically, ≥ 0.8 of the 10 or 5 chunks are required to be covered by reads. If a query has multiple matches from different genomes of the same species, it is still considered a uniquely matched read. Additionally, we require a matched reference genome with some high-confidence uniquely matched reads with high k-mer coverage ($\geq$0.75, approximately equal to $\geq$98% sequence similarity). For the second round, all reads are reassigned, and the matched reference genomes are filtered with the above criteria and two additional criteria, including the maximal standard deviation of relative genome chunk depths and the minimal percentage of high-confidence uniquely matched reads. Additionally, the two-stage taxonomy assignment algorithm in MegaPath (Leung *et al.*, 2020) is used to further reduce the false positives of ambiguous matches. For the third round, all reads are reassigned, and the matched reference genomes are filtered with the same criteria as in round two. Then, the relative abundances of all matched references are estimated with a strategy similar to that of Metalign (LaPierre *et al.*, 2020), in which multi-matched queries are proportionally assigned to reference genomes according to the number of uniquely matched reads. In contrast, the taxonomic abundance of a reference is computed with the cumulative matched bases and normalized by the genome size. We preset six profiling modes for multiple analysis scenarios, including mode 0 for pathogen detection, mode 1 for higher recall, mode 2 for high recall, mode 3 for the default mode, mode 4 for high precision, and mode 5 for higher precision (Supplementary section S2.3).

2.5 Generation of KMCP databases

Although KMCP is efficient and convenient in building custom databases, we also provide three prebuilt databases. For bacteria and archaea, representative GTDB r202 (Parks *et al.*, 2022) species datasets including 47,894 genomes are used to build the KMCP database with the following parameters: $k = 21$, number of genome chunks = 10, chunk overlap = 100 bp, number of hash functions = 1, false-positive rate of Bloom filters = 0.3, and number of index files = 32.

Genome_updater 0.2.5 (https://github.com/pirovc/genome_updater) is used to download fungal and viral genomes from the RefSeq or GenBank database. Four hundred and three fungal genomes from RefSeq r208 are used to construct the fungal database with the same parameters as the GTDB database. For GenBank r246, a maximum of five genomes are retained for each viral TaxId, leaving 27,936 viral genomes to build the viral database in which the genomes are split into five chunks, and a lower false-positive rate of Bloom filters (0.05) is used. A snapshot of NCBI Taxonomy taxdump files on Dec 6, 2021, is used.

3 Results

3.1 The KMCP framework

We have designed and implemented KMCP, a novel k-mer-based taxonomic profiling tool. To reduce the high rate of false positives observed in traditional k-mer-based methods, we introduce genomic positions to k-mers by splitting the reference genomes into chunks (Fig. 1a). We track both the origin and the approximate location in the reference (Fig. 1b) for each query, which is referred to as *pseudo-mapping*. The coverage information of genome chunks help determine the existence of the genome (Dadi *et al.*, 2017) (Fig. 1c). A three-round profiling method is adopted to filter out suspicious matches to increase profiling specificity (methods and Supplementary Fig. S2).

KMCP provides modular subcommands for k-mer computation, index construction, read searching, search result merging, and taxonomic profiling (Supplementary Table S1 and Fig. S3). Due to the simplicity of the COBS index, database construction (including k-mer computation and indexing) is highly efficient. And we have improved the indexing speed (25 vs. 30 minutes for indexing 47,894 prokaryotic genomes from GTDB) and reduced the index size (56 vs. 87 GB, Supplementary Table S2). In addition, we have optimized the search algorithm and achieved a ten-fold increase in the speed of short read batch searching relative to COBS (Supplementary Fig. S4).

Unlike LCA-based methods, which retrieve the TaxId of each k-mer in the read and assign the LCA of the resulting TaxIds to the read, KMCP searches across the database with all k-mers and returns reference genome chunks sharing enough k-mers with the query. The searching step does not assign any taxonomic label to the query reads; therefore, search results from multiple databases can be merged. The mergeability of the search results makes it possible to construct separate databases for different reference genome datasets and choose various databases to search with considerable flexibility. Users can also create many small databases, each with a partition of reference genomes, and accelerate the search using computer clusters (section 3.6). Furthermore, taxonomic information, including NCBI Taxonomy dump files and a plain text file mapping reference identifiers to TaxIds, is only needed in the profiling step, allowing the database to be easily updated with newer taxonomy information.

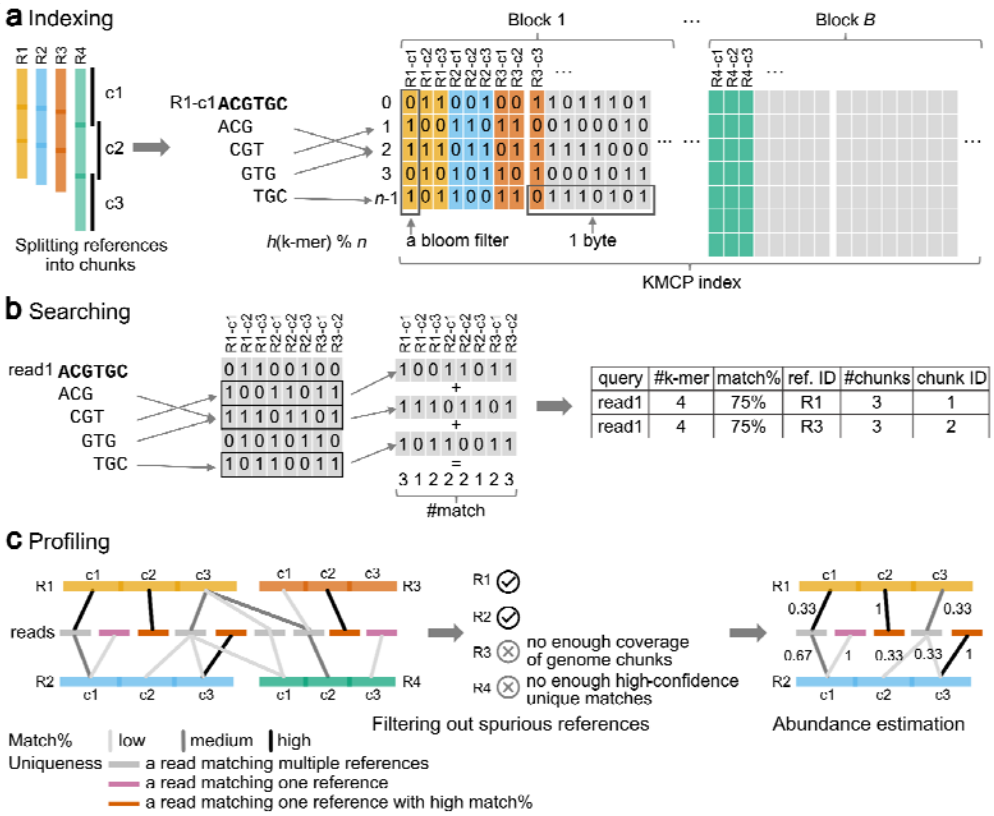

Fig. 1. Schematic representation of KMCP. **(a)** Reference genomes are split into chunks, and the k-mers are stored in a modified COBS index. **(b)** The k-mers of a read are computed and searched in the index, and reference chunks sharing enough matched k-mers with the query are returned. **(c)** Matched references are filtered based on criteria including chunk coverage information and k-mer similarity, and relative abundances are then estimated. Match% refers to the k-mer coverage (the proportion of matched k-mers and distinct k-mers of a query).

3.2 Evaluation based on CAMI marine datasets

We evaluated KMCP against the ten simulated marine datasets from the second round of CAMI challenges compared with the top seven performant metagenomic profiling tools (Meyer *et al.*, 2022), including mOTUs 2.5.1, MetaPhlAn 2.9.22, Centrifuge 1.0.4beta, DUDes 0.08, Metalign 0.6.2, Bracken 2.2, and CCMetagen 1.1.3 (Marcelino *et al.*, 2020). KMCP built a database with 9,871 reference prokaryotic and 8,242 viral genomes, with a maximum of five genomes for each bacterial and archaeal species, from the same Jan 8, 2019, RefSeq snapshot used by other tools (Supplementary section S1.3). The gold standard profiles and results of different tools were downloaded and compared to those of KMCP using OPAL (Meyer *et al.*, 2019).

KMCP showed taxon identification and relative abundance accuracies close to those of mOTUs2 and MetaPhlAn2 at the genus rank, while KMCP outperformed other tools at the species rank (Fig. 2a). The average completeness (recall, 0.915) of KMCP was slightly lower than that of Bracken (0.944) while higher than those of mOTUs2 (0.809), MetaPhlAn2 (0.850), and Centrifuge (0.887) at the species rank. For purity (precision), the value of KMCP (0.830) was lower than these of mOTUs2 (0.887) and DUDes (0.871). For relative abundance measured based on L1 norm error, KMCP had the second-lowest value at the genus rank and the lowest at the

species rank. The weighted UniFrac error (Fig. 2b), which measures the similarity across taxonomic ranks between the gold standard and estimated abundances (Lozupone and Knight, 2005; Meyer *et al.*, 2019), showed that KMCP had the lowest value. When we filtered out the predictions with the lowest relative abundances, totaling 0-1% within the species rank, the purity of all tools increased. However, the completeness decreased because the datasets were designed to show logarithmic abundance distributions with many low-abundance species (Fig. 2c).

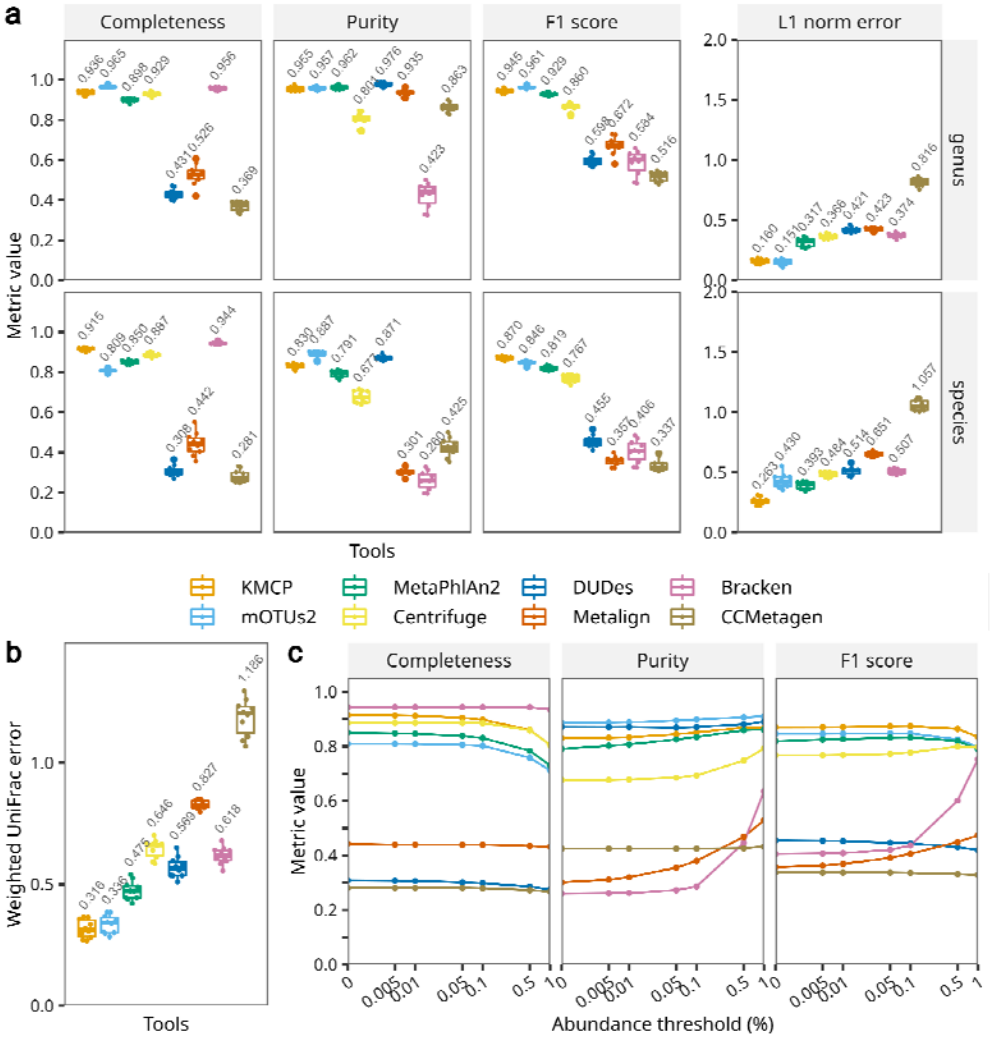

Fig. 2. Taxonomic profiling accuracy based on CAMI mouse gut datasets. **(a)** Assessment of taxon identification accuracy (completeness/recall, purity/precision, and F1 score) and relative abundance accuracy (L1 norm error) at the ranks of genus and species, with no abundance thresholding. **(b)** Relative abundance accuracy across all ranks, with no abundance thresholding. **(c)** Change in taxon identification accuracy at the species rank with abundance thresholds of 0 to 1%.

3.3 Evaluation based on simulated prokaryotic communities

We further evaluated KMCP alongside other taxonomy profilers of the newest versions with the latest databases, including Bracken 2.6.2, Centrifuge 1.0.4, Ganon 1.1.2, DUDes 0.08, SLIMM 0.3.4, MetaPhlAn 3.0.13 (Beghini *et al.*, 2021), and mOTUs 3.0.1. Sun *et al.* (Sun *et al.*, 2021)

simulated twenty-five simulated metagenomic sequence reads from five habitats with genomes selected from the intersection among the reference databases of MetaPhlAn2, mOTUs2, and Kraken2/Bracken (Supplementary section S1.4). Each dataset contained 3 Gb of 150-bp paired-end reads and presented a log-norm-distributed profile. To minimize database bias in the benchmarks, we built databases for Bracken, Bracken, Centrifuge, Ganon, DUDes, and SLIMM using the identical reference genomes in the KMCP database. Since MetaPhlAn3 and mOTUs3 are closely linked to official databases, the most recent embed databases were used.

The k-mer-based methods (KMCP, Bracken, Centrifuge, and Ganon) and marker gene-based methods (MetaPhlAn3 and mOTUs3) showed good completeness, with average values above 0.9 at both the genus and species ranks, whereas MetaPhlAn3 showed a lower value (Fig. 3a), and the alignment-based tools DUDes and SLIMM presented inadequate completeness. In terms of purity, KMCP, MetaPhlAn3, and mOTUs3 exhibited the highest values (above 0.9) at the genus rank, followed by Centrifuge (0.740). At the species rank, the average purity of mOTUs3 remained above 0.9, while those MetaPhlAn3 and KMCP dropped to 0.855 and 0.813, respectively. In terms of the F1 score, KMCP performed better than MetaPhlAn3 and Centrifuge at both the genus and species ranks. For L1 norm error, mOTUs3 performed the best, followed by MetaPhlAn3 and KMCP. The Weighted UniFrac error results showed a similar trend to the L1 norm error at the species rank (Fig. 3b). When low-abundance predictions were filtered out, the purities of Bracken, Centrifuge, and Ganon significantly increased but were still lower than those of mOTUs, MetaPhlAn3, and KMCP (Fig. 3c).

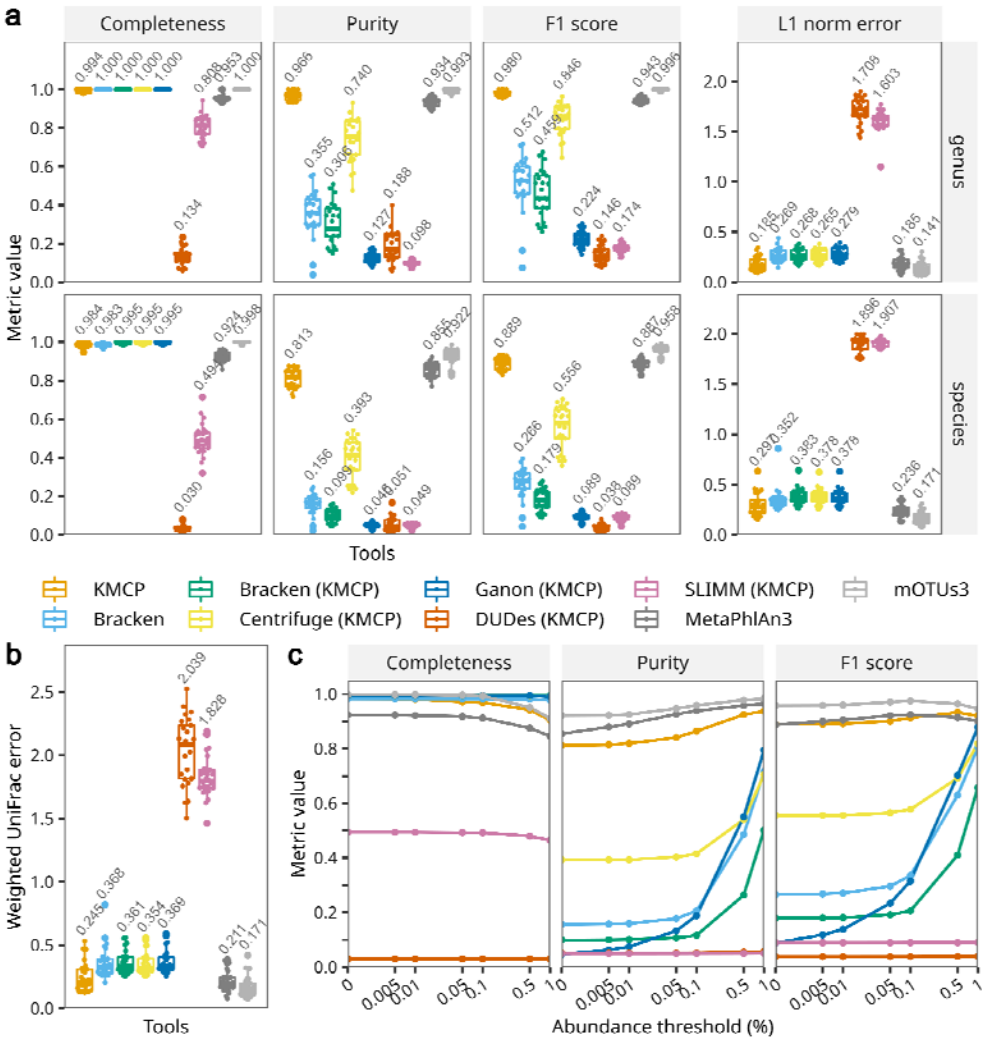

Fig. 3. Taxonomic profiling accuracy based on simulated bacterial communities. Tools whose names are followed by (KMCP) use databases created with the same reference genomes used by KMCP. **(a)** Assessment of taxon identification accuracy (completeness/recall, purity/precision, and F1 score) and relative abundance accuracy (L1 norm error) at the ranks of genus and species, with no abundance thresholding. **(b)** Relative abundance accuracy across all ranks, with no abundance thresholding. **(c)** Change in taxon identification accuracy at the species rank with abundance thresholds of 0 to 1%.

3.4 Evaluation based on mock virome datasets

Roux *et al.* (Roux *et al.*, 2016) designed two mock viral communities, including two ssDNA viruses and ten dsDNA viruses, with 16 datasets generated with three sequencing libraries. The clean data contained 50-250 bp of clean reads, and the total amount of data ranged from 900 Mb to 2.5 Gb. mOTUs3, without virus detection support, was not included in this benchmark. MetaPhlAn3 was run with the option "*--add_viruses*" to add virus detection. KMCP, Bracken, and Centrifuge use databases constructed from identical reference genomes, including genomes of archaea, bacteria, fungi, and viruses (Supplementary section S1.5).

Since the mock viral communities were created by mixing viral capsids obtained from lysates, undigested bacterial DNA might be retained in the sequencing data. However, the relative abundances at the superkingdom rank showed variable results (Fig. 4a). KMCP and MetaPhlAn3 showed higher average viral abundances (99.72% and 99.02%), while the values generated by the other tools were below 80%. Then, we filtered viruses and recomputed the taxonomic profiles. KMCP, Bracken, Centrifuge, and Ganon, with the same reference genomes, showed completeness values higher than 0.9 at both the genus and species ranks (Fig. 4b). Bracken, with the official database, and MetaPhlAn3 presented much lower completeness, although they used databases built with reference genomes obtained after the publication of the sequencing data. For purity, the values of KMCP and Centrifuge were above 0.95 at the genus rank, while they decreased to 0.863 and 0.691, respectively, at the species rank, and the purity values of Bracken (KMCP) decreased from 0.857 to 0.369. Regarding taxon identification, KMCP ranked first at the species rank, with an average F1 score of 0.9, followed by Centrifuge, with a score of 0.814. KMCP also showed a better abundance estimation accuracy across taxonomic ranks (Fig. 4b and 4c).

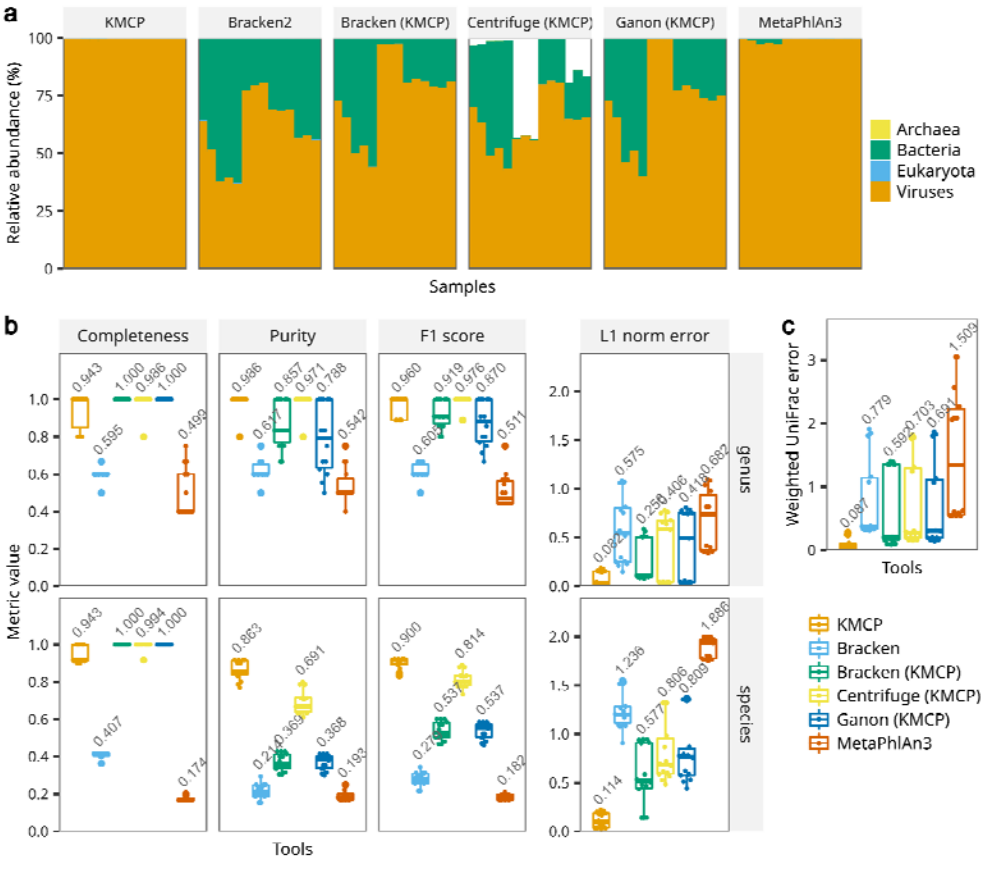

Fig. 4. Taxonomic profiling accuracy based on mock virome communities. **(a)** Unfiltered microbial composition predicted by all the tools. **(b)** Assessment of viral presence-absence (completeness/recall, purity/precision, and F1 score) and relative abundance (L1 norm error) accuracy at the genus and species ranks, with no abundance thresholding. **(c)** Relative abundance accuracy of viruses across ranks, with no abundance thresholding. Tools whose names are followed by (KMCP) use databases constructed with the same reference genomes as KMCP.

3.5 Evaluation of pathogen detection in clinical samples

The above results indicated that KMCP could accurately detect both prokaryotic and viral organisms in simulated and real metagenomic communities. We next evaluated whether KMCP could detect pathogens from low-depth clinical samples. Gu *et al.* (Gu *et al.*, 2021) collected 182 body fluid samples from 160 patients as residual samples after routine clinical testing in a microbiology laboratory; among these samples, 87 with accessible short-read datasets that were verified by culture or 16S rRNA gene qPCR were analyzed with KMCP, Kraken2, Bracken, Centrifuge, and Ganon. Among these 87 samples, seventy-six were positive with one or more pathogens, and eleven were negative. The number of reads in the clean data ranged from 49 to 252,311 (Supplementary Table S3). MetaPhlAn3 and mOTUs3 were not tested in this case due to the limited ability of marker-gene-based methods to detect pathogens with ultralow coverage. According to our results, Bracken generally assigned more reads to the detected pathogens than Kracken2 alone, which did not affect the accuracy. Therefore, we ran Bracken immediately after Kraken2 on all samples. All tools were run according to the previous analysis but with looser thresholds, and KMCP used a preset pathogen detection mode (Supplementary section S1.6).

The results (Table 1) showed that Bracken presented a significantly improved accuracy when using a more extensive database. Bracken, Centrifuge, and Ganon with the KMCP reference genomes all achieved a sensitivity of 80.26%, and KMCP, Bracken, and Ganon all presented a perfect specificity of 100%. The sensitivity of KMCP was slightly lower than those of Bracken, Centrifuge, and Ganon, with one misclassified sample (S19) (Supplementary Table S3). The low specificity of Centrifuge was due to the false-positive detection of a fungus (*Coccidioides immitis*) in three samples (S33, S35, S82).

On the other hand, Bracken, Centrifuge, and Ganon mainly reported the detected organisms and associated read numbers. In contrast, KMCP reported more metrics than the read count, including the genome chunk fraction and similarity score (the 90th percentile of k-mer coverage of all uniquely matched reads), which were also used to sort the predictions in the report (Supplementary section S1.6). These additional metrics allowed KMCP to provide confident results, with fewer predictions (**Fig. 5**a) and a higher mean rank of the pathogens in the prediction list (**Fig. 5**b). For predictions with small numbers of reads, the genome chunk fraction in KMCP results could also provide valuable evidence for determining the existence of the pathogens. For example, predictions with only 4-10 reads could still have genome chunk fractions ranging from 0.2 to 0.5 (**Fig. 5**c).

Table 1. Accuracy of pathogen detection in clinical samples.

Tools	Sensitivity	Specificity
KMCP	78.95%	100%
Bracken	73.68%	81.82%
Bracken (KMCP)	80.26%	100.00%
Centrifuge (KMCP)	80.26%	72.73%
Ganon (KMCP)	80.26%	100.00%

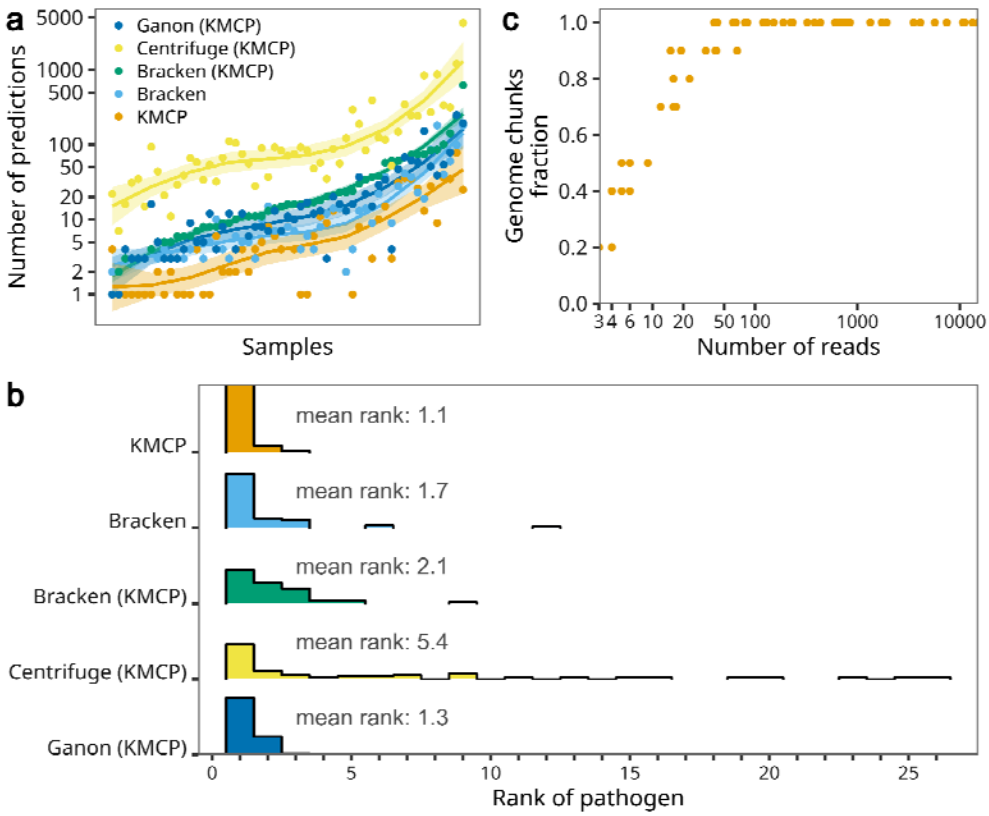

Fig. 5. **(a)** The number of predictions in 55 samples in which all tools correctly detected pathogens. Samples were sorted by the number of predictions with Bracken (KMCP). **(b)** Histogram of pathogen ranks in the 55 taxonomic profiling results. **(c)** The number of reads and the genome chunk fraction in the 60 true-positive predictions of KMCP. Tools whose names are followed by (KMCP) use databases constructed with the same reference genomes as KMCP.

3.6 Benchmarking computational requirements

We measured the runtime and maximum memory usage of database building and taxonomic profiling among KMCP and the above methods (Supplementary section S1.7). Since MetaPhlAn3 and mOTUs3 database construction requires many manual steps (Beghini *et al.*, 2021; Milanese *et al.*, 2019), we only created databases for KMCP, Kraken2, Bracken, Centrifuge, Ganon, DUDes, and SLIMM using 40 threads. We assessed the database building efficiency in terms of database size, building time, and peak memory occupation. The results illustrated that KMCP was much less time-consuming, required less memory than the other methods, and showed a smaller database size than Kraken2, Centrifuge, Ganon, DUDes, and SLIMM (Table 2).

Then, eight samples from the CAMI mouse gut metagenome datasets, each with 16.5 million paired-end 150-bp reads, were used to assess metagenomic profiling time and memory requirements. All tools used 40 threads for taxonomic profiling. The results (Fig. 6) showed that the marker-gene-based methods, including MetaPhlAn3 and mOTUs3, balanced analysis time and memory usage, while the other tools occupied much more memory. KMCP occupied a modest amount of memory (57.3 GB), but its speed was much slower than those of the other tools, with most of the time spent in read searching. Bracken, Centrifuge, and Ganon, which used databases

built with the same reference genomes as KMCP, achieved fast profiling but required more memory, with Bracken occupying more than 250 GB of memory. DUDes and SLIMM both used Bowtie2 for read mapping, which had an index of more than 300 GB (Table 2) and needed more than 200 GB of memory for mapping. The profiling step of DUDes was slower than that of SLIMM. Since KMCP has a scalable searching mode, we ran KMCP in a computer cluster with 24 computation nodes, referring to KMCP_HPC, where each node searched against a small database built with a partition of the reference genomes using 32 threads. The analysis time of KMCP decreased from 533.8 minutes to 23.2 minutes under these conditions.

Table 2. Database size, building time and memory.

Tool	Index size (GB)	Building time	Peak memory (GB)
KMCP	66	21 m	14
Bracken	304	10 h 08 m	258
Centrifuge	97	44 h 35 m	522
Ganon	195	43 m	197
DUDes	318	40 h 48 m	922
SLIMM	317	40 h 34 m	922
MetaPhlAn3	3	/	/
mOTUs3	8	/	/

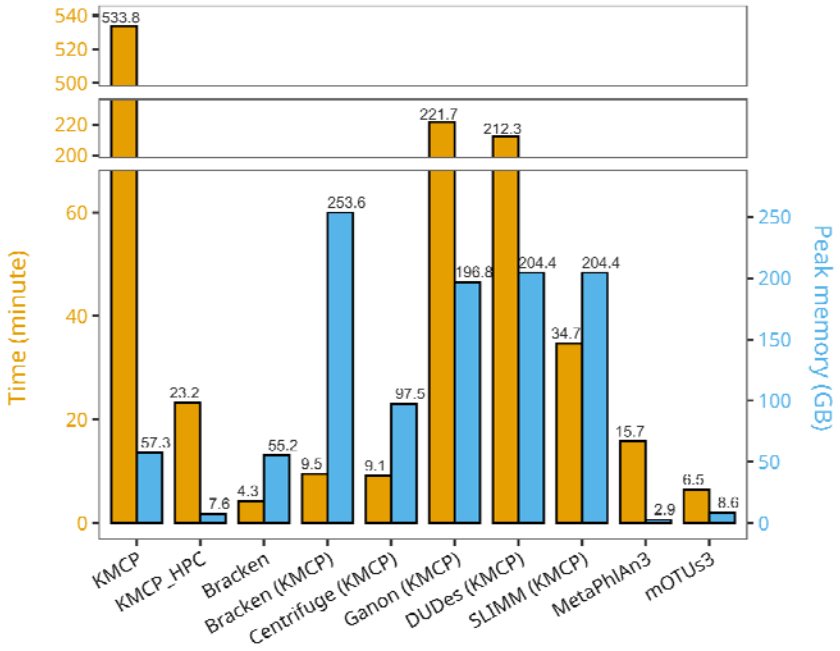

Fig. 6. Per-sample runtime and maximum memory use for metagenomic profiling. KMCP_HPC refers to KMCP running on a computer cluster with each node performing searches against a small database built with a part of the reference genomes. Tools whose names are followed by (KMCP) use databases constructed with the same reference genomes as KMCP.

4 Discussion

The second round of CAMI challenges compared 22 methodological versions, and the results showed that the taxonomic profiling tools performed well among bacterial species. However, the accuracy was lower for archaea and viruses (Meyer *et al.*, 2022), indicating a need for better reference sequence collection and tool development. Most microbiome studies have focused mainly on bacterial, archaeal, and fungal populations, with less attention paid to viral communities. However, bacteriophages play essential roles in microbiome ecology (Liang and Bushman, 2021; Townsend *et al.*, 2021). Current viral metagenomic analysis approaches mainly rely on sequence assembly based on viral metagenomic sequencing data, while sample processing and genome assembly are challenging (Santiago-Rodriguez and Hollister, 2020). With the establishment of an increasing number of human gut prokaryotic (Almeida *et al.*, 2021; Chibani *et al.*, 2022; Hiseni *et al.*, 2021) and virus/phage (Camarillo-Guerrero *et al.*, 2021; Gregory *et al.*, 2020; Nayfach *et al.*, 2021) catalogs, it is possible to employ reference-based taxonomic profiling tools to investigate archaea, bacteria, and viruses simultaneously.

In this context, we have designed and implemented KMCP, a novel k-mer-based taxonomic profiling tool. K-mer-based methods apply fast exact k-mer matching and lack classification sensitivity for distant homologous genomes. Kraken2 uses the spaced seed (Brinda *et al.*, 2015) to increase classification sensitivity. In KMCP, we use a small k value of 21 and retain all matches with k-mer coverage ≥ 0.55 by default, without LCA computation. A k-mer coverage of 0.55, approximately equal to a sequence identity of 96.5%, is chosen to balance the sensitivity of sequence searching and the generation of false positives resulting from the probabilistic data structure. Users can also apply a lower threshold by constructing a database with a lower false-positive rate at the expense of larger index size.

To minimize the false positives resulting from the lower k-mer coverage threshold, we set a threshold for the required proportion of high-confidence uniquely matched reads. Furthermore, to combine the fast speed of k-mer-based methods and position information in alignment methods, we introduce the novel concept of *pseudo-mapping*. By dividing the reference genomes into chunks and indexing k-mer sets alongside the chunk positions, KMCP utilizes the genome coverage (chunks faction) to filter out spurious references. Thus, the reference genome chunks need to be uniformly covered by reads. Additionally, we require a matched genome with some high-confidence uniquely matched reads with high k-mer coverage.

SLIMM also uses reference genome coverage information to remove unlikely genomes (Dadi *et al.*, 2017) while it maps reads into bins of specific width (e.g., 1 kb) after the read mapping step. In contrast, KMCP splits reference genomes into predefined numbers (e.g., 10 for bacteria and fungi) of chunks in database building.

Benchmarking results based on simulated and real data indicate that KMCP accurately performs taxonomic profiling in both prokaryotic and viral populations and provides confident pathogen detection in infectious clinical samples. When applied to the simulated CAMI 2 marine metagenome datasets, KMCP presented the highest average F1 score at the species rank, and its L1 norm error value was the lowest. When applied to the prokaryotic datasets simulated with genomes selected from the intersection among the reference databases of MetaPhlAn2, mOTUs2, and Kraken2/Bracken reported in Sun *et al.*, KMCP presented a slightly higher average F1 score than MetaPhlAn3 at the species rank, while the average L1 norm error was higher than those of MetaPhlAn3 and mOTUs3. Although mOTUs3 showed good accuracy based on prokaryotic metagenome datasets, it lacks the ability to detect viruses. The benchmarking results based on the

real mock virome datasets showed that KMCP outperformed Centrifuge, Bracken, and MetaPhlAn3, with higher accuracies of both taxon identification and abundance estimation.

We also showed that KMCP could detect low-coverage pathogens, including bacteria, fungi, and viruses, with as few as two reads in infectious clinical samples. KMCP presented a specificity of 100% and a sensitivity of 78.95%, which was slightly lower than the 80.26% sensitivity of Kraken2/Bracken and Ganon. In addition, KMCP generated shorter prediction lists and identified pathogens in priority positions by considering both sequence similarity and the genome chunk fraction, which could help researchers rapidly interpret the reports. Furthermore, the database size and memory occupation of KMCP were much lower than those of Kraken2/Bracken, Centrifuge, and Ganon with the same reference genomes.

Extensive and comprehensive reference databases of high quality are essential for producing good metagenomic results, and the increasing number of references imposes challenges regarding database size and construction efficiency among taxonomic profilers (Breitwieser *et al.*, 2019). The searchable data structure BIGSI was invented to efficiently search for plasmid and resistance gene sequences in an index containing 447,833 microbial genomes (Bradley *et al.*, 2019). Later, COBS improved BIGSI in terms of index size and searching speed. KMCP, reported in this paper, reimplements and modifies the COBS data structure. The index size of KMCP is smaller than that of COBS, and the batch searching speed is increased by ten times. However, after splitting the reference genomes into chunks, KMCP presents an $O(nK)$ query time, compared to the $O(K)$ query time of BIGSI/COBS, where K is the number of reference genomes, and n is the number of genome chunks. We also implemented the RAMBO data structure (Gupta *et al.*, 2019) (similar to BIGSI but closer to SBTs), which presents a sublinear query time at the cost of memory space and accuracy. However, the much larger index size did not improve the query efficiency in standalone machines, and the higher false-positive rate of queries negatively impacted the profiling accuracy. Fortunately, KMCP could utilize computer clusters to linearly accelerate read searching, with each node searching against a small database constructed with a partition of the reference genomes.

Ganon and KMCP are both efficient in indexing a large number of reference genomes. IBF, a data structure very similar to BIGSI, is used in Ganon for indexing k-mers. Technically, IBF, BIGSI, and COBS all interleave each row of bits of Bloom filters and save them as a byte list in either database (BIGSI) or a binary file (IBF and COBS). Ganon uses a smaller k-mer size (19) than KMCP (21) by default, theoretically generating fewer k-mers. Ganon sets a low false-positive rate of 0.05 and uses three hash functions for Bloom filters to balance database size and classification accuracy. In KMCP, we use a high false-positive rate of 0.3 for creating smaller databases and use only one hash function to increase search speed. However, the benchmarking results based on both prokaryotic and viral datasets demonstrated that KMCP presents higher accuracies than Ganon in taxon identification, as the latter method shows low precision among low-abundance taxa.

The accuracy of taxonomic profiling relies on the reference genome and the microbial taxonomy. The NCBI taxonomy database is a curated classification and nomenclature database that incorporates all organisms available in major public sequence databases, including GenBank and RefSeq (Schoch *et al.*, 2020). However, with the continuous updating of the taxonomy database, TaxIds are added, deleted, and merged into other TaxIds; the taxonomic ranks, names, and lineage are also changed (Shen and Ren, 2021). The modular design of KMCP separates taxonomic information from the k-mer index, allowing the database to adapt to new taxonomy

versions. As reference genomes, we chose representative species genomes from GTDB, which provides a standardized bacterial and archaeal taxonomy based on genome phylogeny, to build the prokaryotic database. Fungal genomes from RefSeq and viral genomes from GenBank were used to create fungal and viral databases, respectively. Based on the same reference genomes, KMCP generates a much smaller database and requires less memory for searching than Kraken2/Bracken, Centrifuge, and Ganon. Moreover, the lack of genome preprocessing (clustering or low-complexity-region masking) and the fast indexing speed of KMCP enable users to build custom databases efficiently, and scalable searching allows users to perform searches against multiple databases. To update databases, users can either rebuild the database with newly added reference genomes or just build an additional database for the new genomes because search results against multiple databases can be rapidly merged.

There is still room for improvement. For prokaryotic metagenome datasets, KMCP showed a lower purity (precision) at the species rank than at the genus rank, and the abundance estimation was less accurate than those of MetaPhlAn3 and mOTUs3. We consider it worthwhile to improve taxon identification and abundance estimation in future work.

Acknowledgments

We thank Zhi-Luo Deng (Department of Computational Biology, Helmholtz Centre for Infection Research, Braunschweig, Germany) for valuable advice on metagenomic profiling and benchmarking. We thank Robert Clausecker (Zuse Institute Berlin, Germany) for developing the high-performance vectorized positional popcount package (pospop) during the development of KMCP. We thank Jishan Tan (Department of Clinical Laboratory, General Hospital of Western Theater Command, Chengdu, China) for clinical microbiology diagnosis consultations. We thank Chenhao Li (Massachusetts General Hospital, Broad Institute, USA), Zhi-Luo Deng, Qi Zhao (Sun Yat-sen University Cancer Center, Guangzhou, China), and Yujie Zhao for advising and commenting on the manuscript.

Funding

This work was supported by the National Natural Science Foundation of China [32000474 to W.S.]; by the China Postdoctoral Science Foundation [2021M700640 to W.S.]; and by the Chongqing Talents Project [cstc2021ycjh-bgzxm0150 to P.H.].

Conflict of Interest: none declared.

References

Almeida, A., *et al.* (2021) A unified catalog of 204,938 reference genomes from the human gut microbiome. Nat Biotechnol, **39**(1), 105-114.

Beghini, F., *et al.* (2021) Integrating taxonomic, functional, and strain-level profiling of diverse microbial communities with bioBakery 3. Elife, **10**.

Bingmann, T., *et al.* COBS: A Compact Bit-Sliced Signature Index. In. Cham: Springer International

Publishing; 2019. p. 285-303.

Bradley, P., *et al.* (2019) Ultrafast search of all deposited bacterial and viral genomic data. Nat Biotechnol, **37**(2), 152-159.

Breitwieser, F.P., *et al.* (2019) A review of methods and databases for metagenomic classification and assembly. Brief Bioinform, **20**(4), 1125-1136.

Brinda, K., *et al.* (2015) Spaced seeds improve k-mer-based metagenomic classification. Bioinformatics, **31**(22), 3584-3592.

Buchfink, B., *et al.* (2015) Fast and sensitive protein alignment using DIAMOND. Nat Methods, **12**(1), 59-60.

Camarillo-Guerrero, L.F., *et al.* (2021) Massive expansion of human gut bacteriophage diversity. Cell, **184**(4), 1098-1109 e1099.

Caporaso, J.G., *et al.* (2012) Ultra-high-throughput microbial community analysis on the Illumina HiSeq and MiSeq platforms. ISME J, **6**(8), 1621-1624.

Chibani, C.M., *et al.* (2022) A catalogue of 1,167 genomes from the human gut archaeome. Nat Microbiol, **7**(1), 48-61.

Chiu, C.Y. and Miller, S.A. (2019) Clinical metagenomics. Nat Rev Genet, **20**(6), 341-355.

Dadi, T.H., *et al.* (2017) SLIMM: species level identification of microorganisms from metagenomes. PeerJ, **5**, e3138.

Dadi, T.H., *et al.* (2018) DREAM-Yara: an exact read mapper for very large databases with short update time. Bioinformatics, **34**(17), i766-i772.

Gregory, A.C., *et al.* (2020) The Gut Virome Database Reveals Age-Dependent Patterns of Virome Diversity in the Human Gut. Cell Host Microbe, **28**(5), 724-740 e728.

Gu, W., *et al.* (2021) Rapid pathogen detection by metagenomic next-generation sequencing of infected body fluids. Nat Med, **27**(1), 115-124.

Gupta, G., *et al.* RAMBO: Repeated And Merged BloOm Filter for Ultra-fast Multiple Set Membership Testing (MSMT) on Large-Scale Data. In.; 2019. p. arXiv:1910.02611.

Harris, R.S. and Medvedev, P. (2020) Improved representation of sequence bloom trees. Bioinformatics, **36**(3), 721-727.

Hiseni, P., *et al.* (2021) HumGut: a comprehensive human gut prokaryotic genomes collection filtered by metagenome data. Microbiome, **9**(1), 165.

Human Microbiome Project, C. (2012) Structure, function and diversity of the healthy human microbiome. Nature, **486**(7402), 207-214.

Irber, L., *et al.* (2022) Lightweight compositional analysis of metagenomes with FracMinHash and minimum metagenome covers. bioRxiv, 2022.2001.2011.475838.

Kim, D., *et al.* (2016) Centrifuge: rapid and sensitive classification of metagenomic sequences. Genome Res, **26**(12), 1721-1729.

Langmead, B. and Salzberg, S.L. (2012) Fast gapped-read alignment with Bowtie 2. Nat Methods, **9**(4), 357-359.

LaPierre, N., *et al.* (2020) Metalign: efficient alignment-based metagenomic profiling via containment min hash. Genome Biol, **21**(1), 242.

Leung, C.M., *et al.* (2020) MegaPath: sensitive and rapid pathogen detection using metagenomic NGS data. BMC Genomics, **21**(Suppl 6), 500.

Liang, G. and Bushman, F.D. (2021) The human virome: assembly, composition and host interactions. Nat Rev Microbiol, **19**(8), 514-527.

Lozupone, C. and Knight, R. (2005) UniFrac: a new phylogenetic method for comparing microbial communities. Appl Environ Microbiol, **71**(12), 8228-8235.

Lu, J., *et al*. (2017) Bracken: estimating species abundance in metagenomics data. PeerJ Computer Science, **3**, e104.

Marcelino, V.R., *et al.* (2020) CCMetagen: comprehensive and accurate identification of eukaryotes and prokaryotes in metagenomic data. Genome Biol, **21**(1), 103.

Marchet, C., *et al.* (2021) Data structures based on k-mers for querying large collections of sequencing data sets. Genome Res, **31**(1), 1-12.

Menzel, P., *et al.* (2016) Fast and sensitive taxonomic classification for metagenomics with Kaiju. Nat Commun, **7**, 11257.

Meyer, F., *et al.* (2019) Assessing taxonomic metagenome profilers with OPAL. Genome Biol, **20**(1), 51.

Meyer, F., *et al.* (2022) Critical Assessment of Metagenome Interpretation: the second round of challenges. Nat Methods.

Milanese, A., *et al*. (2019) Microbial abundance, activity and population genomic profiling with mOTUs2. Nat Commun, **10**(1), 1014.

Mohamadi, H., *et al*. (2016) ntHash: recursive nucleotide hashing. Bioinformatics, **32**(22), 3492-3494.

Nasko, D.J., *et al.* (2018) RefSeq database growth influences the accuracy of k-mer-based lowest common ancestor species identification. Genome Biol, **19**(1), 165.

Nayfach, S., *et al*. (2021) Metagenomic compendium of 189,680 DNA viruses from the human gut microbiome. Nat Microbiol, **6**(7), 960-970.

O'Leary, N.A., *et al*. (2016) Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. Nucleic Acids Res, **44**(D1), D733-745.

Parks, D.H., *et al*. (2022) GTDB: an ongoing census of bacterial and archaeal diversity through a phylogenetically consistent, rank normalized and complete genome-based taxonomy. Nucleic Acids Res, **50**(D1), D785-D794.

Piro, V.C., *et al*. (2020) ganon: precise metagenomics classification against large and up-to-date sets of reference sequences. Bioinformatics, **36**(Suppl_1), i12-i20.

Piro, V.C., *et al*. (2016) DUDes: a top-down taxonomic profiler for metagenomics. Bioinformatics, **32**(15), 2272-2280.

Quince, C., *et al*. (2017) Shotgun metagenomics, from sampling to analysis. Nat Biotechnol, **35**(9), 833-844.

Roux, S., *et al*. (2016) Towards quantitative viromics for both double-stranded and single-stranded DNA viruses. PeerJ, **4**, e2777.

Santiago-Rodriguez, T.M. and Hollister, E.B. (2020) Potential Applications of Human Viral Metagenomics and Reference Materials: Considerations for Current and Future Viruses. Appl Environ Microbiol, **86**(22).

Sayers, E.W., *et al*. (2022) GenBank. Nucleic Acids Res, **50**(D1), D161-D164.

Schoch, C.L., *et al*. (2020) NCBI Taxonomy: a comprehensive update on curation, resources and tools. Database (Oxford), **2020**.

Sczyrba, A., *et al*. (2017) Critical Assessment of Metagenome Interpretation-a benchmark of metagenomics software. Nat Methods, **14**(11), 1063-1071.

Shen, W. and Ren, H. (2021) TaxonKit: A practical and efficient NCBI taxonomy toolkit. J Genet Genomics, **48**(9), 844-850.

Solomon, B. and Kingsford, C. (2016) Fast search of thousands of short-read sequencing experiments. Nat Biotechnol, **34**(3), 300-302.

Solomon, B. and Kingsford, C. (2018) Improved Search of Large Transcriptomic Sequencing Databases Using Split Sequence Bloom Trees. J Comput Biol, **25**(7), 755-765.

Sun, C., *et al.* (2018) AllSome Sequence Bloom Trees. J Comput Biol, **25**(5), 467-479.

Sun, Z., *et al*. (2021) Challenges in benchmarking metagenomic profilers. Nat Methods, **18**(6), 618-626.

Townsend, E.M., *et al*. (2021) The Human Gut Phageome: Origins and Roles in the Human Gut Microbiome. Front Cell Infect Microbiol, **11**, 643214.

Truong, D.T., *et al.* (2015) MetaPhlAn2 for enhanced metagenomic taxonomic profiling. Nat Methods, **12**(10), 902-903.

Wood, D.E., *et al*. (2019) Improved metagenomic analysis with Kraken 2. Genome Biol, **20**(1), 257.

Ye, S.H., *et al.* (2019) Benchmarking Metagenomics Tools for Taxonomic Classification. Cell, **178**(4), 779-794.