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] Institute for Viral Hepatitis, Department of Infectious Diseases, Key Laboratory of Molecular Biology for Infectious Diseases, Chinese Ministry of Education, The Second Affiliated Hospital of Chongqing Medical University, 74 Linjiang Road, Yuzhong District, Chongqing, 400010, China

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

Abstract

A growing number of microbial reference genomes enable better metagenomic profiling accuracy yet put higher requirements on the indexing efficiency, database size, and runtime of taxonomic profilers. Besides, most profilers focused mainly on bacterial, archaeal, and fungal populations with less attention on viral communities. We present KMCP, a novel k-mer based metagenomic profiling tool that introduces genomic positions to k-mers by splitting the reference genomes into chunks. Benchmarking results on both simulated and real data demonstrate that KMCP not only allows for accurate taxonomic profiling of archaea, bacteria, and viral populations from metagenomic shotgun sequence data, but also provides confident pathogen detection for infectious clinical samples of low depth. KMCP is implemented in Go and is available as open-source software, under MIT, at https://github.com/shenwei356/kmcp.

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

Background

High-throughput sequencing technologies and computational methods have been combined into shotgun metagenomics approaches to study the composition and dynamics of microbial communities composed of bacteria, archaea, viruses, and fungi [1]. Metagenomics is an essential means to explore microbial diversity [2], the interaction between microorganisms and hosts [3], and the rapid detection of uncultivated and difficult-to-cultivate pathogens [4]. Metagenomic sequencing is mainly based on the next-generation high-throughput sequencing platform, where each sample can generate millions of 50-250 bp short reads. 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 [5].

Several studies have assessed metagenomic profiling tools for taxonomic classification and profiling [5-8]. Based on the methodology, these tools can be categorized into three types: (1) k-mer (length-k subsequences) based, such as Kraken2 [9], Bracken [10], and Centrifuge [11]; (2) alignment-based, such as DIAMOND [12], and Kaiju [13], DUDes [14], and SLIMM [15]; (3) marker-gene based, including MetaPhlAn2 [16] and mOTUs2 [17]. 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. And the taxonomic assignment algorithm based on the Lowest Common Ancestor (LCA) often has low accuracy at low taxonomic ranks due to closely related species sharing a large fraction of k-mers, and further decreasing with the increasing number of reference genomes [18]. Alignment-based methods have a high tolerance for base variation; however, they are computationally expensive with larger database sizes and relatively slow searching speed, which make it not suitable for time-sensitive scenarios such as detecting pathogenic microorganisms in clinical samples. Marker gene approaches perform well on the detection of archaea and bacteria [7], while it is difficult to detect viruses accurately with this strategy since viruses do not have universally conserved genes such as the 16S and 18S rRNA genes [13].

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 [18, 19]. Databases such as NCBI RefSeq [20] and Genbank [21] have increasing amounts of microbial genomes, e.g., 250,000 assemblies belonging to 48,000 species in RefSeq release 210, which pose a big challenge to the taxonomic profilers' database size and construction efficiency. Recently, several genome catalogs have been created, aiming to provide comprehensive and high-quality reference genomes for taxonomic and functional research, such as the human gut prokaryotic genomes collections (UHGG [22], HumGut [23], and an archaea catalog [24]) and human gut virus catalogs (GVD [25], GPD [26], MGV [27]). Sourmash [28] outperformed other tools on the Critical Assessment of Metagenome Interpretation (CAMI) mouse gut datasets [29] using FracMinHash sketches with a scale of 10,000 calculated from all the 141,677 prokaryotic genomes in a RefSeq snapshot. The sketching algorithm uses a small number of signatures from the whole genome sequences; therefore, it could index a large number of reference genomes. However, it is limited with reduced sensitivity on small genomes such as viruses.

Several computational approaches have been introduced to index and query large collections of sequencing data sets and genomes over the last few years [30]. K-mer aggregative methods index each data set using a separate Bloom filter, then build an aggregation data structure for querying.

Sequence Bloom Tree (SBT) approaches [31-34] are designed for collections with high k-mer redundancy, such as human RNA-seq sequencing data. In contrast, Bloom filter matrices methods, including BItsliced Genomic Signature Index (BIGSI) [35] and Compact Bit-Sliced Signature Index (COBS) [36], are suitable for heterogeneous k-mer sets, such as 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. Though 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 [31, 36]. These features indicate that k-mer based Bloom filter matrices methods can be applied to index reference genomes for taxonomic profiling.

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 produce a low classification resolution, usually above species rank. An alignment method, SLIMM [15], performs read mapping with Bowtie2 [37] and maps matched reads into bins of fixed-width across the reference genome; then, it utilizes coverage information of reference genomes to remove unlikely genomes and gain more uniquely mapped reads to assign 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), which 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 on simulated and real metagenomic datasets comprised of archaea, bacteria, and viruses, and the benchmarking results indicate that KMCP performs well on both prokaryotic and viral populations.

Results

The KMCP framework

We have designed and implemented KMCP, a novel k-mer based taxonomic profiling tool. To reduce the high false positives in the 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, denoted as *pseudo-mapping*. The split genome chunks can reduce the false-positive of k-mer matching by narrowing the searching space, and the coverage information of genome chunks can help determine the existence of the genome [15] (Fig. 1c). A three-round profiling method is adopted to filter out suspicious matches for increasing profiling specificity (methods and Fig. S1).

KMCP provides modular sub-commands for k-mer computation, index construction, reads searching, search results merging, and taxonomic profiling (Table S1). Thanks to the simplicity of the COBS index, the database construction (including k-mer computation and indexing) is highly efficient, taking only 25 minutes for indexing 47,894 prokaryotic genomes from GTDB (Table S2). And we have improved the indexing speed and reduced the index size (Table S2). Besides, we take much effort to optimize the searching algorithm and gain near ten times speedup for short reads batch searching compared to COBS (Fig. S2). Unlike LCA-based methods, which retrieve TaxId for 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 querying reads; therefore, search results from multiple databases can be merged. The mergeability of search results makes it flexible to construct separate databases for different reference genome datasets and choose various databases to search. Users could also create many small databases, each with a partition of the reference genomes, and accelerate the searching using computer clusters (see "Benchmarking computational requirements" section). Furthermore, the taxonomic information, including NCBI Taxonomy dump files and a plain text file mapping reference identifiers to TaxIds, are only needed in the profiling step, allowing the easy update of the database to newer taxonomy information.

We provide three pre-built databases for metagenomic profiling (see Methods), including the prokaryotic database created with 47,894 archaea and bacteria from GTDB r202 [38], the viral database with 27,936 virus genomes from Genbank r246, and the fungal database with 403 fungi genomes from RefSeq r208. The prokaryotic and fungal genomes are split into ten chunks, and the viral genomes are five due to the smaller genome sizes (see Methods). To obtain a proper searching threshold that takes both sensitivity and false-positive rate into account, we analyzed the correlation between k-mer similarity and sequence identity. And a k-mer coverage (proportion of matched k-mers and distinct k-mers of a query) cutoff of 0.55 (approximately equal to the sequence identity of 96.5%) was selected, with accepted false-positive rates for short-read metagenomics profiling (Fig. S3).

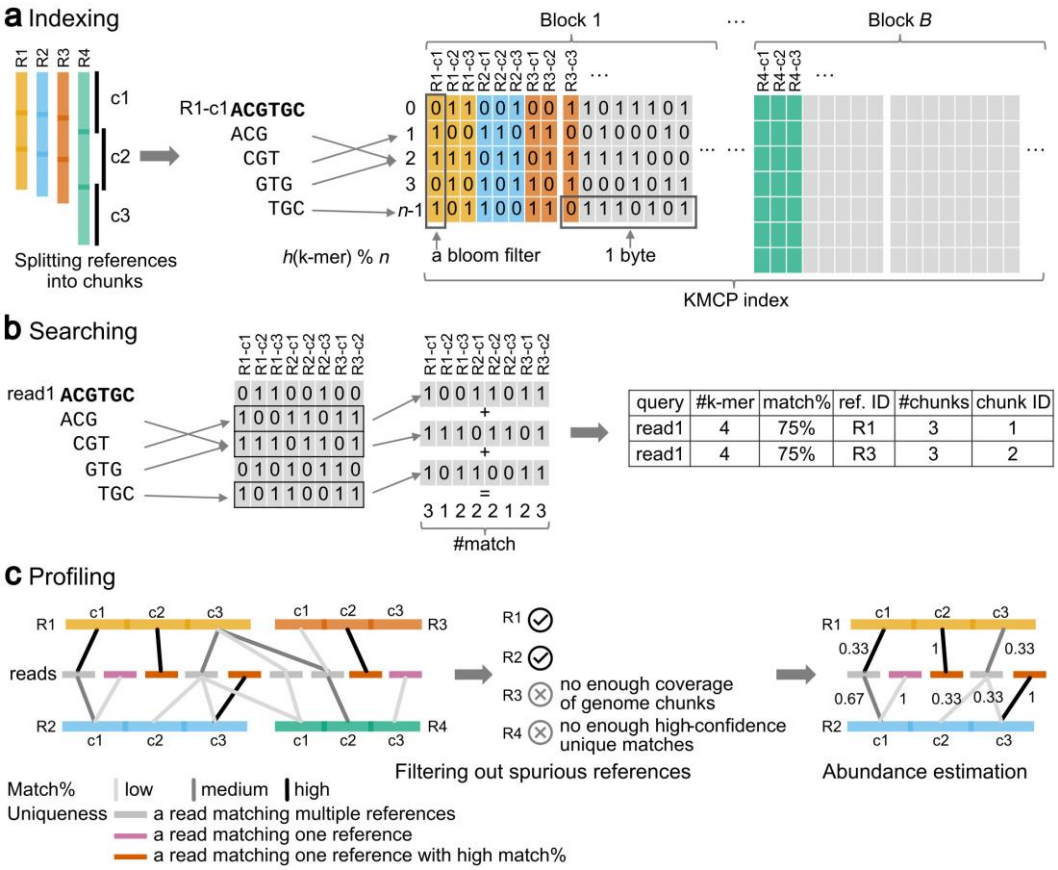

Figure 1. Schematic of KMCP. **a)** References genomes are split into chunks, and the k-mers are stored in a modified COBS index. **b)** K-mers of a read are computed and searched in the index, and reference chunks sharing enough matched k-mers with the query are returned. **c)** Matched references are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

mers coverage (the proportion of matched k-mers and distinct k-mers of a query).

Evaluation on CAMI mouse gut datasets

We evaluated KMCP on the 64 simulated mouse gut datasets from the second round of CAMI challenges, comparing with the top five performant metagenomic profiling tools [39], including Bracken, MetaPalette [40], MetaPhyler [41], MetaPhlAn2, and mOTUs2. KMCP built a database with 9,871 reference prokaryotic and 8,242 viral genomes, with a maximum of five genomes for each bacteria and archaea species, from the same Jan 8, 2019, RefSeq snapshot used by other tools (see methods). The gold standard profiles and results of different tools were downloaded and compared to KMCP using OPAL [29].

KMCP showed a close taxon identification and relative abundance accuracy to MetaPhlAn2 and mOTUs2 at the genus and species rank (Fig. 2a). The average completeness (recall) of KMCP was lower than both MetaPhlAn2 and mOTUs2 at the two ranks. And the average F1 score of KMCP was higher than MetaPhlAn2 while lower than mOTUs2 at both ranks. For the relative abundance measured by L1 norm error, KMCP was the lowest at the genus rank while ranked the second one after MetaPhlAn2 at the species rank. The Weighted UniFrac error (Fig. 2b), measuring the similarity across taxonomic ranks between the gold standard and estimated abundances [29, 42], showed mOTUs2 performed best and KMCP was the second-best one. When filtering out the predictions with the smallest relative abundances summing up to 0-1% within the species rank, the purity (precision) of all tools increased. Meantime, the completeness decreased significantly because the datasets were designed to have logarithmic abundance distributions with many low-abundance species (Fig. 2c).

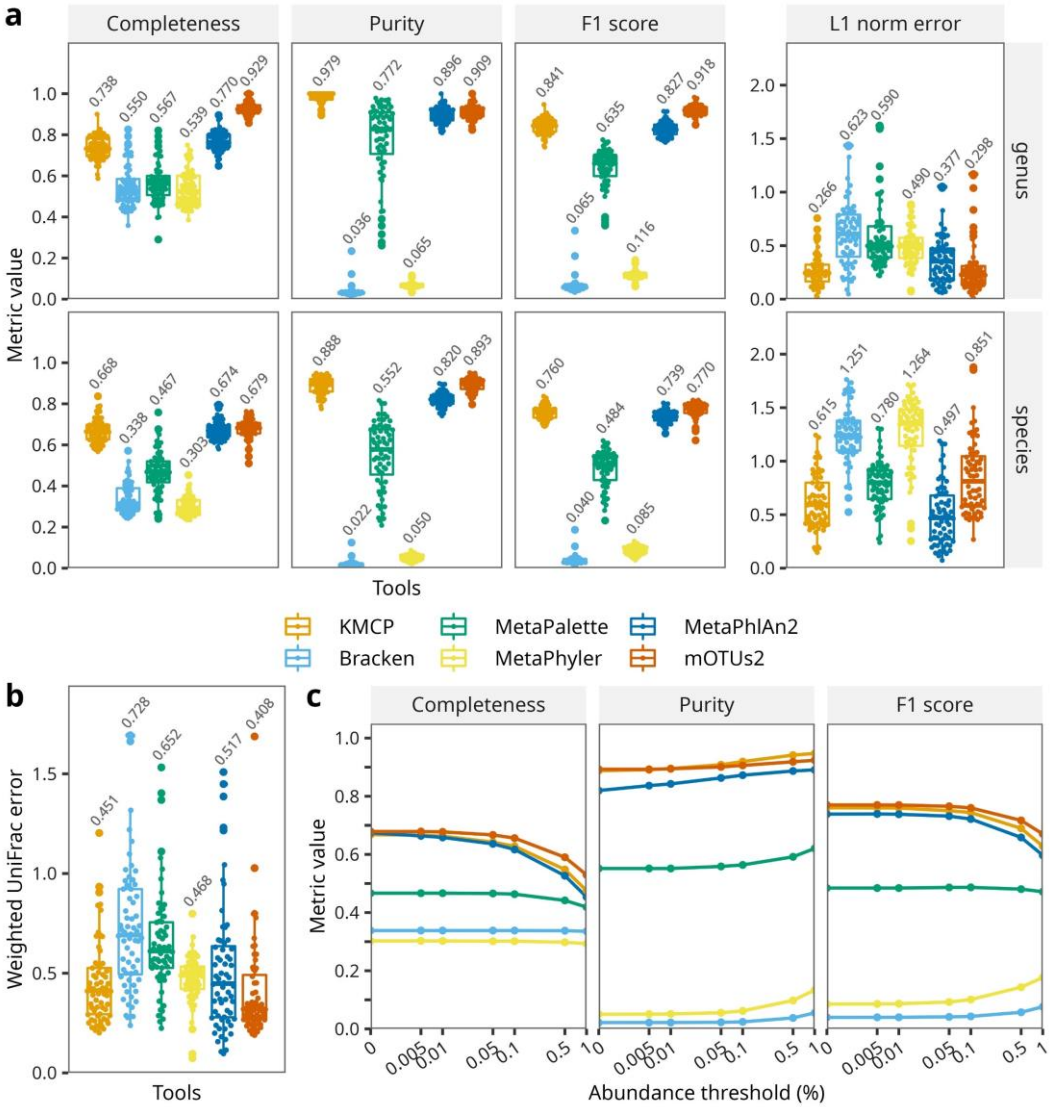

Figure 2. Taxonomic profiling accuracy on CAMI mouse gut datasets. **a)** Assessment of taxon identification accuracy (completeness/recall, purity/precision, F1 score) and relative abundance accuracy (L1 norm error) at the rank of genus and species, with no abundance thresholding. **b)** Relative abundance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

Evaluation on simulated prokaryotic communities

Then we evaluated KMCP alongside other taxonomy profilers, including Bracken 2.6.2, Centrifuge 1.0.4, DUDes 0.08, SLIMM 0.3.4, MetaPhlAn 3.0.13[43], and mOTUs 3.0.1. Sun *et al*. [6] 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. Each dataset contained 3 Gb 150-bp paired-end reads and had a log-norm distributed profile. To minimize the database bias in the benchmarks, we built databases for Bracken, Bracken, Centrifuge, DUDes, SLIMM, using the identical reference genomes in the database of KMCP. Since MetaPhlAn3 and mOTUs3 are tight with the official databases, the latest databases were used.

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

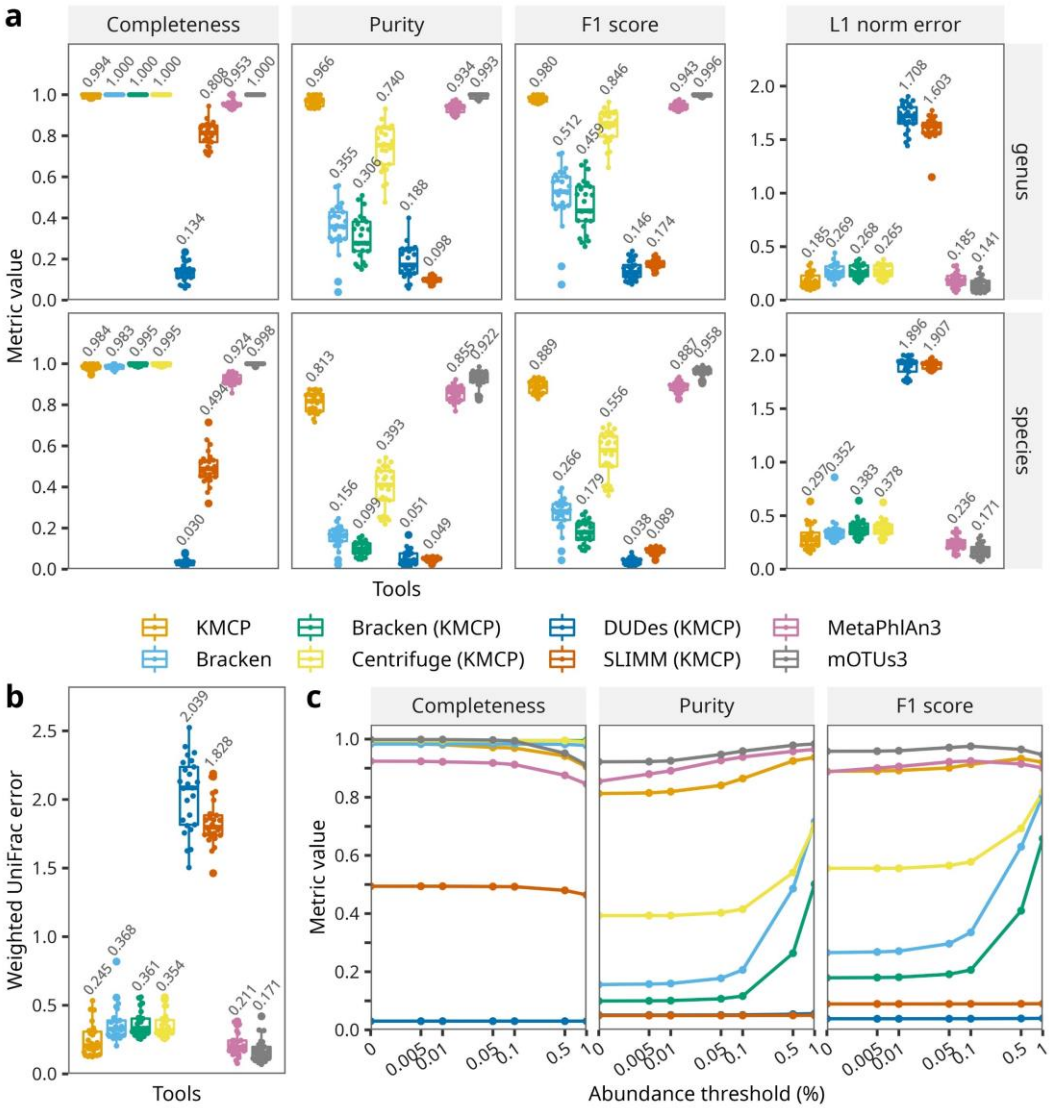

Figure 3. Taxonomic profiling accuracy on simulated bacteria communities. Tools with a "(KMCP)" suffix means they used databases created with the same reference genomes as KMCP. **a)** Assessment of taxon identification accuracy (completeness/recall, purity/precision, F1 score) and relative abundance accuracy (L1 norm error) at the rank 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 species rank with abundance thresholds from 0 to 1%.

Evaluation on mock virome datasets

Roux *et al*. [44] designed two mock viral communities, including two ssDNA and ten dsDNA viruses, with 16 datasets generated with three sequencing libraries. The clean data contain 50-250 bp clean reads, and the data sizes range from 900 Mb to 2.5 Gb. The mOTUs3 without the support of virus detection was not included in this benchmark. And MetaPhlAn3 was run with the option "*—add_viruses*" for adding detection of viruses. KMCP, Bracken, and Centrifuge used databases constructed from identical reference genomes, including archaea, bacteria, fungi, and viruses.

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

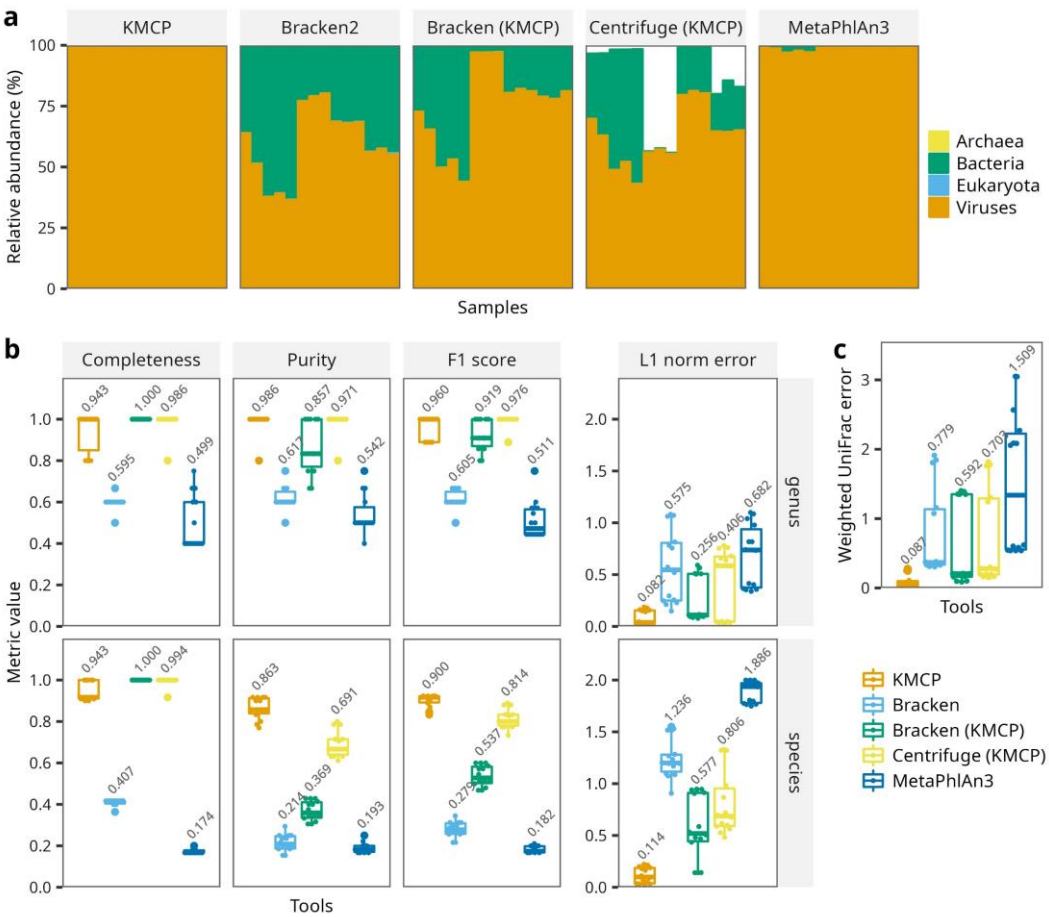

Figure 4. Taxonomic profiling accuracy on mock virome communities. **a)** Unfiltered microbial composition predicted by all the tools. **b)** Assessment of viral presence-absence (completeness/recall, purity/precision, F1 score) and relative abundance (Bray-Curtis distance) accuracy at rank genus and species, with no abundance thresholding. **c)** Relative abundance accuracy of viruses across ranks, with no abundance thresholding. Tools with a "(KMCP)" prefix denote they use databases constructed with the same reference genomes as KMCP.

Evaluation of pathogen detection in clinical samples

The above results indicated that KMCP could accurately detect both prokaryotic and viral organisms at simulated and real metagenomic communities. We next evaluated whether KMCP could detect pathogens from low-depth clinical samples. Gu *et al*. [45] collected 182 body fluid samples from 160 patients as residual samples after routine clinical testing in the microbiology laboratory, of which 87 with accessible short-read datasets and verified by culture or 16S rRNA gene qPCR were analyzed with KMCP, Kraken2, Bracken, and Centrifuge. Of the 87 samples, seventy were positive with one or more pathogens, and eleven were negative. The number of reads in the clean data ranges from 49 to 252,311 (Table S3). MetaPhlAn3 and mOTUs3 were not tested due to the limited ability of marker-gene-based methods to detect pathogens with ultra-low coverage. In our observation, Bracken generally assigned more reads to the detected pathogens than Kracken2 alone, which did not affect the accuracy. So we ran Bracken right after Kraken2 for all samples. All tools ran following previous analysis but with looser thresholds, and KMCP used a preset pathogen detection mode (see method).

The result (Table 1) showed that Bracken significantly improved the accuracy using a more extensive database. Bracken and Centrigue with KMCP's reference genomes both achieved a

sensitivity of 87.14%, and KMCP and Bracken both had a perfect specificity of 100%. The sensitivity of KMCP was slightly lower than Bracken and Centrifuge, with one misclassified sample (S19) (Table S3). The low specificity of Centrifuge was due to the false-positively detection of a fungus (*Coccidioides immitis*) in three samples (S33, S35, S82).

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

Table 1. Accuracy of pathogen detection in clinical samples.

	Sensitivity	Specificity
KMCP	85.71%	100%
Bracken	80.00%	81.82%
Bracken (KMCP)	87.14%	100.00%
Centrifuge (KMCP)	87.14%	72.73%

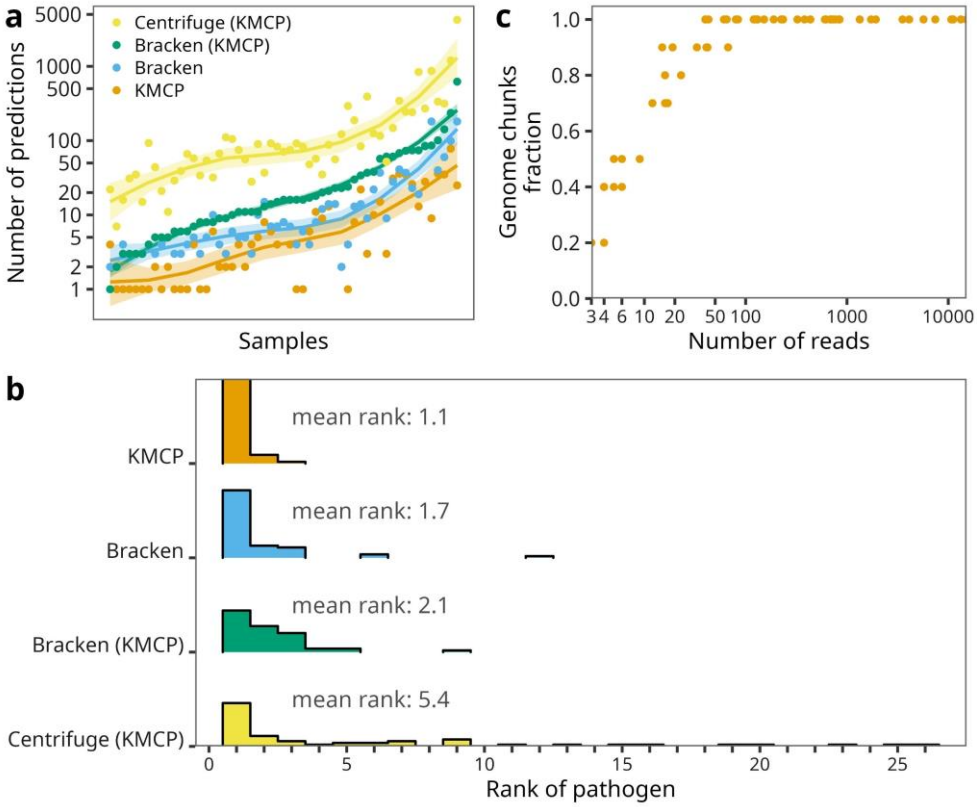

Figure 5. a) The number of predictions in 55 samples in which all tools correctly detected pathogens. Samples were sorted by the number of predictions with Bracken (KMCP). **b)** Histogram of pathogen ranks in the 55 taxonomic profiling results. **c)** The number of reads and the genome chunks fraction in the 60 true-positive predictions of KMCP. Tools with a "(KMCP)" suffix mean they use databases constructed with identical reference genomes as KMCP.

Benchmarking computational requirements

We measured the runtime and maximum memory usage of database building and taxonomic profiling for KMCP and the above methods. Since the database construction of MetaPhlAn3 and mOTUs3 require many manual steps [17, 43], we only created databases for KMCP, Kraken2, Bracken, Centrifuge, DUDes and SLIMM using 40 threads. We assessed the database building efficiency in terms of database size, building time, and peak memory occupation. The result illustrated that KMCP was much less time-consuming, required less memory than others, and held a smaller database size than Kraken2, Centrifuge, 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 the metagenomic profiling time and memory requirement. All tools used 40 threads for taxonomic profiling. The results (Fig. 6) showed that the marker-genes-based methods, including MetaPhlAn3 and mOTUs3, balanced analysis time and memory usage, while other tools occupied much more memory. KMCP occupied a modest memory (57.3GB), but the speed was much slower than other tools, spending most of the time in reads searching. Bracken and Centrifuge, using databases built with the identical reference genomes as KMCP, were fast in profiling but needed more memories, of which Bracken occupied more than 250 GB memory. DUDes and SLIMM both used Bowtie2 for reads mapping, which had an index of more than 300 GB (Table 2) and needed more than 200 GB memory for mapping. And the profiling step of DUDes was slower than SLIMM. Since KMCP had 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 part of the reference genomes using 32 threads. Then the analysis time of KMCP decreased from 533.8 minutes to 23.2 minutes.

Table 2. Index database size and 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
DUDes	318	40 h 48 m	922
SLIMM	317	40 h 34 m	922
MetaPhlAn3	3	/	/
mOTUs3	8	/	/

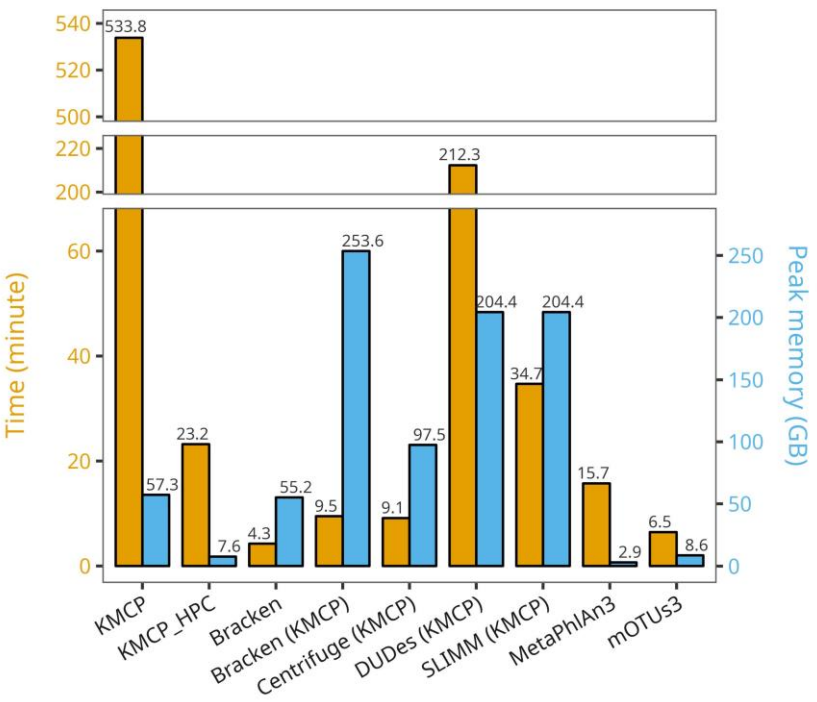

Figure 6. Per sample runtime and maximum memory usage for metagenomic profiling. KMCP_HPC refers to KMCP running on a computer cluster with each node performing searching against a small database built with a part of the reference genomes. Tools with a "(KMCP)" suffix mean they use databases constructed with identical reference genomes as KMCP.

Discussion

The second round of CAMI challenges compared 22 method versions, and the results showed taxonomic profiling tools performing well in bacterial species. At the same time, the accuracy was lower for archaea and viruses [7], indicating a need for better reference sequence collection and tool development. Most microbiome studies focused mainly on bacterial, archaeal, and fungal populations with less attention on viral communities. However, bacteriophages have essential roles in the microbiome ecology [46, 47]. Current viral metagenomic analysis approaches mainly rely on sequence assembly from viral metagenomic sequencing data, while it is challenging in sample processing and genome assembly [48]. With the establishment of the increasing number of human gut prokaryotic [22-24] and virus/phage [25-27] catalogs, it is possible for reference-based taxonomic profiling tools to investigate archaea, bacteria, and viruses simultaneously.

We have designed and implemented KMCP, a novel k-mer based taxonomic profiling tool. K-mer based methods apply fast k-mer exact matching, lacking classification sensitivity to distant homologous genomes. Kraken2 uses the spaced seed [49] to increase classification sensitivities. In KMCP, we use a small k value of 21 and keep all matches with k-mer coverage ≥ 0.55 by default, with no LCA computation. The k-mer coverage 0.55, approximately equal to the sequence identity of 96.5%, is chosen to balance the sensitivity of sequence searching and false positivity brought by the probabilistic data structure. Users can also use a lower threshold by constructing a database with a lower false-positive rate at the expense of larger index size.

To minimize the false positive brought by the lower k-mer coverage threshold, we set a threshold

for requiring a proportion of the high-confidence uniquely matched reads. Furthermore, to combine the fast speed of k-mer based methods and position information in alignment methods, we introduce a novel concept, *pseudo-mapping*. By splitting the reference genomes into chunks and indexing k-mers sets alongside the chunk positions, KMCP utilizes the genome coverage (chunks faction) to filter out spurious references. In other words, the reference genome chunks need to be uniformly covered by reads, of which there should exist some proportion of reads with high similarity.

SLIMM also uses coverage information of reference genomes to remove unlikely genomes [15], while it maps reads into bins of specific width (e.g., 1 Kb) after the reads mapping step. Differently, KMCP splits reference genomes into predefined numbers (e.g., 10 for bacteria and fungi) of chunks in database building.

Benchmarking results 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. On the simulated CAMI mouse gut metagenome datasets, KMCP had an average F1 score close to that of the two marker-gene based tools MetaPhlAn2 and mOTUs2 at species rank, and the value of L1 norm error was the second-lowest. On the prokaryotic datasets simulated with genomes selected from the intersection among the reference databases of MetaPhlAn2, mOTUs2, and Kraken2/Bracken in Sun *et al*. [6], KMCP had a slightly higher average F1 score than MetaPhlAn3 at species rank, while the average L1 norm error was higher than that of MetaPhlAn3 and mOTUs3. Though mOTUs3 has good accuracy on prokaryotic metagenome datasets, it lacks the ability of viral detection. The benchmarking results on the real mock virome datasets showed that KMCP outperformed Centrifuge, Bracken, and MetaPhlAn3, with high 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 had a specificity of 100% and a sensitivity of 85.71%, slightly lower than the 87.14% of the Kraken2/Bracken. Besides, KMCP generated shorter prediction lists and showed pathogens in priority positions by considering both sequence similarity and the genome chunks fraction, which could help researchers rapidly interpret the reports. Furthermore, the database size and memory occupation of KMCP are much smaller than Kraken2/Braken, and Centrifuge with the same reference genomes.

Extensive and comprehensive reference databases of high quality are essential to producing good metagenomic results, and the increasing number of references brings challenges to database size and construction efficiency to taxonomic profilers [19]. A searchable data structure named BIGSI was invented to efficiently search for plasmid and resistance genes sequences in an index containing 447,833 microbial genomes [35]. Later, COBS improved the BIGSI in index size and the searching speed. In this paper, KMCP re-implemented and modified the COBS data structure. The index size of KMCP was smaller than COBS, and the batch searching speed was increased by near ten times. However, after splitting the reference genomes into chunks, KMCP has an $O(nK)$ query time compared to $O(K)$ 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 [50], similar to BIGSI but closer to SBTs, with sub-linear query time at the cost of memory spaces and accuracy. However, the much bigger index size did not improve 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 the reads searching, with each node searching against a small database constructed with a partition of the reference genomes.

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 [51]. However, with the continuous update of the taxonomy database, TaxIds are added, deleted, and merged into others; meanwhile, the taxonomic ranks, names, and lineage are also changed [52]. The modular design of KMCP separates taxonomic information from the k-mer index, allowing the database to adapt to new taxonomy versions. For reference genomes, we chose the species representative genomes from GTDB, a standardized bacterial and archaea 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. With the identical reference genomes, KMCP generates a much smaller database and requires less memory for searching than Kraken2/Bracken and Centrifuge. Moreover, requiring no preprocessing of genomes (clustering nor low-complexity region masking) and the fast indexing speed of KMCP enables users to build custom databases efficiently, and the scalable search allows users to perform searches against multiple databases. To update databases, users can either re-built the database with newly added reference genomes or only build an additional database for the new genomes because search results against multiple databases can be rapidly merged.

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

Conclusions

KMCP is a novel taxonomic profiling tool that estimates the relative abundance of archaea, bacteria, viruses, and fungi at the species level for metagenomic shotgun sequence data. KMCP introduces genomic positions to k-mers by splitting the reference genomes into chunks, then computes and stores nucleotide k-mers of reference genomes in a modified and optimized COBS index for fast alignment-free sequence searching. 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 on simulated and real data demonstrated that KMCP is performant on both prokaryotic and viral populations and provides confident pathogen detection for infectious clinical samples. KMCP is implemented in Go, and the source code is freely available under the MIT License. The latest version of KMCP can be obtained from https://github.com/shenwei356/kmcp and the Bioconda channel [53] of Conda package manager.

Methods

Indexing

KMCP efficiently builds a database from a collection of genomes 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 to reduce the false positive rate of k-mer based sequence searching; and the k-mers' location information are further utilized in taxonomic profiling. For genomes without a single complete genome sequence, the chromosomes or contigs are concatenated

with intervals of *k*-1 bases of N to avoid introducing fake k-mers. For each genome, non-redundant nucleotide canonical k-mers (*k*=21, by default) are hashed into 64-bit unsigned integers with the ntHash [54]. 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 column-wise concatenated 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 biggest genome chunk. To avoid generating extremely large Bloom filters for the last block with the biggest 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. Besides, the index files of all blocks are not concatenated to a big 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.

Searching

For searching sequences against the KMCP database, the k-mers are computed with the parameters of the database and sent to query against Bloom filters of all index files in parallel. Like COBS, index files are mapped into virtual address space using *mmap* by default or loaded into memory. Though COBS has a good querying performance compared 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 advantages 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 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. But the array lookup needs 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 column-wise 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) genomes chunks in one function invoking. Besides, sixty-four is the cache line size for most 64-bit computers, which helps to reduce cache miss and brings extra speedup. For the left k-mers (<64) of the query sequence, we unroll the loops of 63 possible k-mer sizes for higher 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 Bloom filters as 0.3 and use one hash function by default following BIGSI and COBS. During searching, the threshold of k-mers coverage (proportion of matched k-mers and

unique k-mers of a query) is set to 0.55 (approximately equal to the sequence identity of 96.5%, Fig. S3d), which keeps the false-positive probability of a query at a reasonable level according to the formula (5) in [31], e.g., 0.003 for 150-bp reads with 83 k-mers matched and 0.028 for 100-bp reads with 55 k-mer matched.

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. Besides, 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 searching speed is linearly related to the number of reference genome chunks in the databases and 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 with each computation node searching 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.

Profiling

In the indexing phase, reference genomes are split into 10 or 5 chunks, which reduce the size of the k-mer matching collection and therefore reduce the false-positive rate of a query sequence. And the coverage information of all chunks can help determine the existence of a genome, thus reduce false positives [15]. To implement this strategy, a three-round profiling method is applied (Fig. S1). For the first round, reference genomes with at least one uniquely matched read are collected and filtered by the genome chunks 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. Meanwhile, we require a matched reference genome having some high-confidence uniquely matched reads with high k-mer coverages (≥0.75, approximately equal to ≥98% sequence similarity). For the second round, all reads are reassigned, and the matched reference genomes are filtered with the above 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 [55] 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 round two. Then the relative abundances of all matched references are estimated with a similar strategy as Metalign [56], in which multi-matched queries are proportionally assigned to reference genomes according to the number of uniquely matched reads. Differently, the taxonomic abundance of a reference is computed with the accumulative 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.

Generation of the KMCP databases

Though KMCP is efficient and convenient for building custom databases, we provide three pre-built databases. For bacteria and archaea, GTDB r202 species representative datasets with 47,894 genomes are used to build KMCP database with parameters of $k = 21$, 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. The genome_updater 0.2.5 (https://github.com/pirovc/genome_updater) is used to download fungi and viral genomes from RefSeq or Genbank database. Four hundred and three fungi genomes from RefSeq r208 are used to construct the fungi 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 for building 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. The snapshot of NCBI Taxonomy taxdump files on Dec 6, 2021, is used.

Benchmarking KMCP using the CAMI framework

Sixty-four simulated short-read mouse gut metagenomic datasets from the CAMI challenge [39], with 5 Gb 150-bp paired-end reads per sample, are used to evaluate the accuracy of KMCP 0.8.0. And the snapshots of RefSeq and NCBI Taxonomy dump files on Jan 08, 2019, are used to create the KMCP database. TaxIds of archaea and bacteria are retrieved with TaxonKit 0.10.1 [52] and used to extract genomes from the RefSeq snapshot. The assembly accessions of all genomes are sorted in alphabet order, and a maximum of five genomes of a species are retained. Since the RefSeq snapshot only contains prokaryotic genomes, we downloaded the viral assembly summary file from RefSeq FTP on Nov 11, 2021, and extracted viral references before Jan 08, 2019. At last, the 9871 prokaryotic and 8,242 viral genomes are used to build KMCP databases with the parameters as described in "Generation of the KMCP databases" section. The gold standards and benchmark results of Bracken 2.5, MetaPhlAn 2.9.21, and mOTUs 2.5.1 are downloaded from https://zenodo.org/communities/cami. OPAL (commit edb6f41 in dev branch) [29] is used to assess taxonomic metagenome profilers. R 4.1.2 [57], ggplot2 3.3.5 [58], ggbreak 0.0.9 [59], and some other R packages are used for data visualization.

Benchmarking KMCP using simulated bacterial communities

Sun *et al*. [6] 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. Each dataset contains 3 Gb 150-bp paired-end reads. KMCP 0.8.0 with the standard database (Dec 6, 2021), Kraken 2.1.2 and Bracken 2.6.2 with the PlusPF collection of the official databases (May 17, 2021), MetaPhlAn 3.0.13 with the database mpa_v30_CHOCOPhlAn_201901 (July 22, 2020), mOTUs 3.0.1 with database 2.6.0 (June 16, 2021) are used for benchmarking. Besides, the genomes for building the KMCP database, including 47,894 prokaryotic genomes from GTDB, 27,936 viral genomes from Genbank, and 403 fungi genomes from RefSeq, are used to create databases for Kraken2/Bracken, Bracken, Centrifuge 1.0.4, DUDes 0.08, and SLIMM 0.3.4 with the default parameters. SeqKit 2.1.0 [60] is used in reference genome preparation.

KMCP uses command of "*kmcp search -j 40 -d $db read1.fq.gz read2.fq.gz -o read.kmcp@$db.tsv.gz*" for searching reads against prokaryotic and fungal databases, command of "*kmcp merge read.kmcp@*.tsv.gz -o read.kmcp.tsv.gz*" for merging search results, and command of "*kmcp profile -m 3 -X $taxdump -T $taxidmap read.kmcp.tsv.gz -o read.kmcp.tsv.gz.k-m3.profile -C read.kmcp.tsv.gz.c-m3.profile --show-rank superkingdom,phylum,class,order,family,genus,species -s $sampleID*" for profiling. MetaPhlAn3 uses command of "*metaphlan --nproc 40 -x $db --bowtie2db $dbdir --input_type fastq read1.fq.gz,read2.right.fq.gz --bowtie2out bowtie2.bz2 --CAMI_format_output -o mpa3.profile*", and mOTUs3 uses command of "*motus profile -t 40 -f*

read1.fq.gz -r read2.fq.gz -p -C precision -o motus.profile". Kraken2 uses command of "*kraken2 --threads 40 --db $db --memory-mapping --gzip-compressed --paired read1.fq.gz read2.fq.gz --report read.kreport > /dev/null*" for classification, and Bracken uses command of "*for r in S G F O C P D; do est_abundance.py -k $db/database150mers.kmer_distrib -l $r -t 10 -i read.kreport -o read.bracken.level-$r; done*" to estimate abundance at species, genus, family, order, class, phylum, and superkingdom rank, then the script tocami.py from CAMI (https://github.com/hzi-bifo/cami2_pipelines/blob/master/bin/tocami.py, commit a0e2ec2) is used to convert the concatenated abundance tables to CAMI taxonomic profiling format. Centrifuge uses command of "*centrifuge -t -p 40 --mm -x $db -q -1 read1.fq.gz -2 read2.fq.gz --report-file read.cf-report.tsv -S read.cf.tsv*" for classification, and uses command of "*centrifuge-kreport -x $db read.cf.tsv > read.kreport*" to generate Kraken-style report, which is further converted to CAMI format with the tocami.py script. DUDes and SLIMM use the same command of "*bowtie2 --mm -p 40 -x $bowtie2db --no-unal --very-fast -k 100 -q -1 read1.fq.gz -2 read2.fq.gz -S read.sam*" for reads mapping. While DUDes uses command of "*DUDes.py -t 1 -s read.sam -d $db -o read*" and SLIMM uses command of "*slimm -w 1000 -o read $db read.sam*" for profiling. The taxonomic profiles of gold standards and all taxonomic profilers are converted to CAMI format with NCBI taxonomy dump file of Dec 6, 2021, and filtered out viruses predictions using TaxonKit 0.10.1.

Benchmarking KMCP using mock virome communities

Roux *et al*. [44] designed two mock viral communities, including two ssDNA and ten dsDNA viruses. Each virus was grown on its specific host (*Pseudoalteromonas*, *Cellulophaga baltica,* or *Escherichia coli*), and viral capsids were obtained from lysates and mixed. Sixteen datasets were generated with three sequencing libraries. The clean data contain paired and unpaired reads with lengths from 50 to 250 bp, and the data sizes of each sample range from 900 Mb to 2.5 Gb. The normalized coverage of each genome was used as a proxy for the relative abundance of each viral genome [44]. The mOTUs3 without the support of virus detection is not included in this benchmark. MetaPhlAn3 was run with the flag "*--add_viruses*" for adding detection of viruses. KMCP searches against the prokaryotic, fungal, and viral databases. After the initial community composition analysis, non-viral predictions (phage hosts) are filtered out using TaxonKit 0.10.1.

Benchmarking KMCP using infectious clinical samples

Gu *et al*. [45] collected 182 body fluid samples from 160 patients as residual samples after routine clinical testing in the microbiology laboratory, of which 87 with accessible short-read datasets and verified by culture or 16S rRNA gene qPCR are available. The number of reads in the clean data ranges from 49 to 252,311 (table S3). Kraken2 and Bracken, Centrifuge, and KMCP are used to detect pathogens, and all tools require a minimal number of reads of two. Differently, KMCP creates databases for prokaryotic genomes from GTDB and fungal genomes from Refseq with a lower false-positive rate of 0.1 rather than 0.3 and uses two hash functions. A small query coverage threshold of 0.4 rather than 0.55 is used for higher sensitivity in the searching step. KMCP uses a preset profiling mode with the highest sensitivity for pathogen detection, in which the predictions are sorted in reverse order by the product of genome chunks fraction and the similarity score, rather than the reads number nor taxonomic abundance in other tools. The score is the 90th percentile of k-mer coverages of all uniquely matched reads. For Bracken, the number of new estimated reads (column *new_est_reads*) is used for sorting. And the number of reads (column *numReads*) is used for Centrifuge.

Benchmarking runtime and storage requirement of database building and profiling

The performance of KMCP and other tools are analyzed on a server with four Intel(R) Xeon(R) Gold 6248 processors and one terabyte main memory, and the reference genomes and databases are stored in solid-state disks while the sequencing reads are saved in hard disk drives. The genomes for building the KMCP databases, including 47,894 prokaryotic genomes from GTDB, 27,936 viral genomes from Genbank, and 403 fungi genomes from RefSeq, are used to create databases for Kraken, Bracken, and Centrifuge with the default parameters. While MetaPhlAn3 and mOTUs3 are tight with the official databases, the latest databases are used. Eight samples from the CAMI mouse gut metagenome datasets are used to assess the analysis time and memory requirement. All tools use 40 threads for database building and taxonomic profiling. The scaling test of KMCP is performed on a computer cluster with 24 nodes, each with one Hygon C86 7185 32-core processor and 124 GB memory. A script (https://github.com/shenwei356/easy_sbatch) is used to batch submitting searching jobs against all databases.

Supplementary Information

Acknowledgments

We thank Zhi-Luo Deng (Department of Computational Biology, Helmholtz Centre for Infection Research, Braunschweig, Germany) for giving 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.

Authors' contributions

W.S. conceived the study and developed KMCP. W.S. analyzed the data and interpreted the results with contributions from H.X., T.H., H.T., M.P., and D.C. W.S., P.H, and H.R. wrote the manuscript. All authors read and approved the manuscript.

Funding

This research was funded by the Natural Science Foundation of China (32000474 to W.S.), the China Postdoctoral Science Foundation (2021M700640 to W.S.), and the Chongqing Talents Project (cstc2021ycjh-bgzxm0150 to P.H.).

Availability of data and materials

The CAMI2 mouse gut datasets and Jan 8, 2019, RefSeq snapshot can be downloaded from https://data.cami-challenge.org/participate, and the gold standard taxonomic profile and profiles of other tools can be downloaded from

https://zenodo.org/search?page=1&size=20&q=keywords:%22CAMI%202%20Mouse%20Gut%20Toy%20data%20set%22&keywords=taxonomic%20profiling. The reference genomes and database used by KMCP can be downloaded from https://1drv.ms/u/s!Ag89cZ8NYcqtjVVADr8r--fnKFt-?e=ivNZNK. Simulated metagenomic sequence reads from Sun *et al*. [6] can be downloaded from https://figshare.com/projects/Pitfalls_and_Opportunities_in_Benchmarking_Metagenomic_Classifiers/79916, and the gold standard profiles in CAMI format can be downloaded from https://github.com/shenwei356/sun2021-cami-profiles/releases/tag/v2021-12-06. Mock virome communities A and B from Roux *et al*. [44] can be downloaded from https://datacommons.cyverse.org/browse/iplant/home/shared/iVirus/DNA_Viromes_library_comparison, and the gold standard profiles in CAMI format can be downloaded from https://github.com/shenwei356/roux2016-mock-virome-cami-profile/releases/tag/v2021-12-06.
The metagenomic samples of infected body fluids from Gu *et al*. [45] can be downloaded from https://www.ncbi.nlm.nih.gov/bioproject/?term=PRJNA558701, and the list of run accession can be downloaded from https://github.com/shenwei356/kmcp/blob/main/benchmarks/real-pathogen-gu2020/acc2sample.tsv.
The source code of KMCP is under MIT license and publicly available on Github https://github.com/shenwei356/kmcp. The prebuilt databases can be downloaded from https://1drv.ms/u/s!Ag89cZ8NYcqtjHwpe0ND3SUEhyrp?e=QDRbEC.

Competing interests

None declared.

References

1. Quince C, Walker AW, Simpson JT, Loman NJ, Segata N: **Shotgun metagenomics, from sampling to analysis.** *Nat Biotechnol* 2017, **35:**833-844.
2. Caporaso JG, Lauber CL, Walters WA, Berg-Lyons D, Huntley J, Fierer N, Owens SM, Betley J, Fraser L, Bauer M, et al: **Ultra-high-throughput microbial community analysis on the Illumina HiSeq and MiSeq platforms.** *ISME J* 2012, **6:**1621-1624.
3. Human Microbiome Project C: **Structure, function and diversity of the healthy human microbiome.** *Nature* 2012, **486:**207-214.
4. Chiu CY, Miller SA: **Clinical metagenomics.** *Nat Rev Genet* 2019, **20:**341-355.
5. Sczyrba A, Hofmann P, Belmann P, Koslicki D, Janssen S, Droge J, Gregor I, Majda S, Fiedler J, Dahms E, et al: **Critical Assessment of Metagenome Interpretation-a benchmark of metagenomics software.** *Nat Methods* 2017, **14:**1063-1071.
6. Sun Z, Huang S, Zhang M, Zhu Q-Y, Haiminen N, Carrieri A-P, Vázquez-Baeza Y, Parida L, Kim H-C, Knight R, Liu Y-Y: **Challenges in Benchmarking Metagenomic Profilers.** *bioRxiv* 2020**:**2020.2011.2014.382994.
7. Meyer F, Fritz A, Deng Z-L, Koslicki D, Gurevich A, Robertson G, Alser M, Antipov D, Beghini F, Bertrand D, et al: **Critical Assessment of Metagenome Interpretation - the second round of challenges.** *bioRxiv* 2021**:**2021.2007.2012.451567.
8. Ye SH, Siddle KJ, Park DJ, Sabeti PC: **Benchmarking Metagenomics Tools for Taxonomic Classification.** *Cell* 2019, **178:**779-794.
9. Wood DE, Lu J, Langmead B: **Improved metagenomic analysis with Kraken 2.** *Genome Biol*

2019, **20:**257.

10. Lu J, Breitwieser FP, Thielen P, Salzberg SL: **Bracken: estimating species abundance in metagenomics data.** *PeerJ Computer Science* 2017, **3:**e104.
11. Kim D, Song L, Breitwieser FP, Salzberg SL: **Centrifuge: rapid and sensitive classification of metagenomic sequences.** *Genome Res* 2016, **26:**1721-1729.
12. Buchfink B, Xie C, Huson DH: **Fast and sensitive protein alignment using DIAMOND.** *Nat Methods* 2015, **12:**59-60.
13. Menzel P, Ng KL, Krogh A: **Fast and sensitive taxonomic classification for metagenomics with Kaiju.** *Nat Commun* 2016, **7:**11257.
14. Piro VC, Lindner MS, Renard BY: **DUDes: a top-down taxonomic profiler for metagenomics.** *Bioinformatics* 2016, **32:**2272-2280.
15. Dadi TH, Renard BY, Wieler LH, Semmler T, Reinert K: **SLIMM: species level identification of microorganisms from metagenomes.** *PeerJ* 2017, **5:**e3138.
16. Truong DT, Franzosa EA, Tickle TL, Scholz M, Weingart G, Pasolli E, Tett A, Huttenhower C, Segata N: **MetaPhlAn2 for enhanced metagenomic taxonomic profiling.** *Nat Methods* 2015, **12:**902-903.
17. Milanese A, Mende DR, Paoli L, Salazar G, Ruscheweyh HJ, Cuenca M, Hingamp P, Alves R, Costea PI, Coelho LP, et al: **Microbial abundance, activity and population genomic profiling with mOTUs2.** *Nat Commun* 2019, **10:**1014.
18. Nasko DJ, Koren S, Phillippy AM, Treangen TJ: **RefSeq database growth influences the accuracy of k-mer-based lowest common ancestor species identification.** *Genome Biol* 2018, **19:**165.
19. Breitwieser FP, Lu J, Salzberg SL: **A review of methods and databases for metagenomic classification and assembly.** *Brief Bioinform* 2019, **20:**1125-1136.
20. O'Leary NA, Wright MW, Brister JR, Ciufo S, Haddad D, McVeigh R, Rajput B, Robbertse B, Smith-White B, Ako-Adjei D, et al: **Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation.** *Nucleic Acids Res* 2016, **44:**D733-745.
21. Sayers EW, Cavanaugh M, Clark K, Pruitt KD, Schoch CL, Sherry ST, Karsch-Mizrachi I: **GenBank.** *Nucleic Acids Res* 2022, **50:**D161-D164.
22. Almeida A, Nayfach S, Boland M, Strozzi F, Beracochea M, Shi ZJ, Pollard KS, Sakharova E, Parks DH, Hugenholtz P, et al: **A unified catalog of 204,938 reference genomes from the human gut microbiome.** *Nat Biotechnol* 2021, **39:**105-114.
23. Hiseni P, Rudi K, Wilson RC, Hegge FT, Snipen L: **HumGut: a comprehensive human gut prokaryotic genomes collection filtered by metagenome data.** *Microbiome* 2021, **9:**165.
24. Chibani CM, Mahnert A, Borrel G, Almeida A, Werner A, Brugere JF, Gribaldo S, Finn RD, Schmitz RA, Moissl-Eichinger C: **A catalogue of 1,167 genomes from the human gut archaeome.** *Nat Microbiol* 2022, **7:**48-61.
25. Gregory AC, Zablocki O, Zayed AA, Howell A, Bolduc B, Sullivan MB: **The Gut Virome Database Reveals Age-Dependent Patterns of Virome Diversity in the Human Gut.** *Cell Host Microbe* 2020, **28:**724-740 e728.
26. Camarillo-Guerrero LF, Almeida A, Rangel-Pineros G, Finn RD, Lawley TD: **Massive expansion of human gut bacteriophage diversity.** *Cell* 2021, **184:**1098-1109 e1099.
27. Nayfach S, Paez-Espino D, Call L, Low SJ, Sberro H, Ivanova NN, Proal AD, Fischbach MA, Bhatt AS, Hugenholtz P, Kyrpides NC: **Metagenomic compendium of 189,680 DNA viruses from the**

human gut microbiome. *Nat Microbiol* 2021, **6:**960-970.

28. Irber L, Brooks PT, Reiter T, Pierce-Ward NT, Hera MR, Koslicki D, Brown CT: **Lightweight compositional analysis of metagenomes with FracMinHash and minimum metagenome covers.** *bioRxiv* 2022**:**2022.2001.2011.475838.
29. Meyer F, Bremges A, Belmann P, Janssen S, McHardy AC, Koslicki D: **Assessing taxonomic metagenome profilers with OPAL.** *Genome Biol* 2019, **20:**51.
30. Marchet C, Boucher C, Puglisi SJ, Medvedev P, Salson M, Chikhi R: **Data structures based on k-mers for querying large collections of sequencing data sets.** *Genome Res* 2021, **31:**1-12.
31. Solomon B, Kingsford C: **Fast search of thousands of short-read sequencing experiments.** *Nat Biotechnol* 2016, **34:**300-302.
32. Sun C, Harris RS, Chikhi R, Medvedev P: **AllSome Sequence Bloom Trees.** *J Comput Biol* 2018, **25:**467-479.
33. Solomon B, Kingsford C: **Improved Search of Large Transcriptomic Sequencing Databases Using Split Sequence Bloom Trees.** *J Comput Biol* 2018, **25:**755-765.
34. Harris RS, Medvedev P: **Improved representation of sequence bloom trees.** *Bioinformatics* 2020, **36:**721-727.
35. Bradley P, den Bakker HC, Rocha EPC, McVean G, Iqbal Z: **Ultrafast search of all deposited bacterial and viral genomic data.** *Nat Biotechnol* 2019, **37:**152-159.
36. Bingmann T, Bradley P, Gauger F, Iqbal Z: **COBS: A Compact Bit-Sliced Signature Index.** In*; Cham*. Springer International Publishing; 2019: 285-303.
37. Langmead B, Salzberg SL: **Fast gapped-read alignment with Bowtie 2.** *Nat Methods* 2012, **9:**357-359.
38. Parks DH, Chuvochina M, Rinke C, Mussig AJ, Chaumeil PA, Hugenholtz P: **GTDB: an ongoing census of bacterial and archaeal diversity through a phylogenetically consistent, rank normalized and complete genome-based taxonomy.** *Nucleic Acids Res* 2022, **50:**D785-D794.
39. Meyer F, Lesker TR, Koslicki D, Fritz A, Gurevich A, Darling AE, Sczyrba A, Bremges A, McHardy AC: **Tutorial: assessing metagenomics software with the CAMI benchmarking toolkit.** *Nat Protoc* 2021, **16:**1785-1801.
40. Koslicki D, Falush D: **MetaPalette: a k-mer Painting Approach for Metagenomic Taxonomic Profiling and Quantification of Novel Strain Variation.** *mSystems* 2016, **1**.
41. Liu B, Gibbons T, Ghodsi M, Treangen T, Pop M: **Accurate and fast estimation of taxonomic profiles from metagenomic shotgun sequences.** *BMC Genomics* 2011, **12 Suppl 2:**S4.
42. Lozupone C, Knight R: **UniFrac: a new phylogenetic method for comparing microbial communities.** *Appl Environ Microbiol* 2005, **71:**8228-8235.
43. Beghini F, McIver LJ, Blanco-Miguez A, Dubois L, Asnicar F, Maharjan S, Mailyan A, Manghi P, Scholz M, Thomas AM, et al: **Integrating taxonomic, functional, and strain-level profiling of diverse microbial communities with bioBakery 3.** *Elife* 2021, **10**.
44. Roux S, Solonenko NE, Dang VT, Poulos BT, Schwenck SM, Goldsmith DB, Coleman ML, Breitbart M, Sullivan MB: **Towards quantitative viromics for both double-stranded and single-stranded DNA viruses.** *PeerJ* 2016, **4:**e2777.
45. Gu W, Deng X, Lee M, Sucu YD, Arevalo S, Stryke D, Federman S, Gopez A, Reyes K, Zorn K, et al: **Rapid pathogen detection by metagenomic next-generation sequencing of infected body fluids.** *Nat Med* 2021, **27:**115-124.
46. Townsend EM, Kelly L, Muscatt G, Box JD, Hargraves N, Lilley D, Jameson E: **The Human Gut**

Phageome: Origins and Roles in the Human Gut Microbiome. *Front Cell Infect Microbiol* 2021, **11:**643214.

47. Liang G, Bushman FD: **The human virome: assembly, composition and host interactions.** *Nat Rev Microbiol* 2021, **19:**514-527.
48. Santiago-Rodriguez TM, Hollister EB: **Potential Applications of Human Viral Metagenomics and Reference Materials: Considerations for Current and Future Viruses.** *Appl Environ Microbiol* 2020, **86**.
49. Brinda K, Sykulski M, Kucherov G: **Spaced seeds improve k-mer-based metagenomic classification.** *Bioinformatics* 2015, **31:**3584-3592.
50. Gupta G, Yan M, Coleman B, Elworth RAL, Medini T, Treangen T, Shrivastava A: **RAMBO: Repeated And Merged BloOm Filter for Ultra-fast Multiple Set Membership Testing (MSMT) on Large-Scale Data.** pp. arXiv:1910.02611; 2019:arXiv:1910.02611.
51. Schoch CL, Ciufo S, Domrachev M, Hotton CL, Kannan S, Khovanskaya R, Leipe D, McVeigh R, O'Neill K, Robbertse B, et al: **NCBI Taxonomy: a comprehensive update on curation, resources and tools.** *Database (Oxford)* 2020, **2020**.
52. Shen W, Ren H: **TaxonKit: A practical and efficient NCBI taxonomy toolkit.** *J Genet Genomics* 2021, **48:**844-850.
53. Gruning B, Dale R, Sjodin A, Chapman BA, Rowe J, Tomkins-Tinch CH, Valieris R, Koster J, Bioconda T: **Bioconda: sustainable and comprehensive software distribution for the life sciences.** *Nat Methods* 2018, **15:**475-476.
54. Mohamadi H, Chu J, Vandervalk BP, Birol I: **ntHash: recursive nucleotide hashing.** *Bioinformatics* 2016, **32:**3492-3494.
55. Leung CM, Li D, Xin Y, Law WC, Zhang Y, Ting HF, Luo R, Lam TW: **MegaPath: sensitive and rapid pathogen detection using metagenomic NGS data.** *BMC Genomics* 2020, **21:**500.
56. LaPierre N, Alser M, Eskin E, Koslicki D, Mangul S: **Metalign: efficient alignment-based metagenomic profiling via containment min hash.** *Genome Biol* 2020, **21:**242.
57. R Core Team: **R: A Language and Environment for Statistical Computing.** R Foundation for Statistical Computing; 2022.
58. Wickham H: *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York; 2016.
59. Xu S, Chen M, Feng T, Zhan L, Zhou L, Yu G: **Use ggbreak to Effectively Utilize Plotting Space to Deal With Large Datasets and Outliers.** *Front Genet* 2021, **12:**774846.
60. Shen W, Le S, Li Y, Hu F: **SeqKit: A Cross-Platform and Ultrafast Toolkit for FASTA/Q File Manipulation.** *PLoS One* 2016, **11:**e0163962.