Bioinformatics, 2024, **40(4)**, btae150
https://doi.org/10.1093/bioinformatics/btae150
Advance Access Publication Date: 16 March 2024
Original Paper

Sequence analysis

CONSULT-II: accurate taxonomic identification and profiling using locality-sensitive hashing

Ali Osman Berk Şapcı [1], Eleonora Rachtman [1], Siavash Mirarab [1,2,]*

[1]Bioinformatics and Systems Biology Graduate Program, University of California, San Diego, CA 92093, United States
[2]Department of Electrical and Computer Engineering, University of California, San Diego, CA 92093, United States

*Corresponding author. Department of Electrical and Computer Engineering, University of California, San Diego, CA 92093, United States.
E-mail: smirarab@ucsd.edu (S.M.)

Associate Editor: Yann Ponty

Abstract

Motivation: Taxonomic classification of short reads and taxonomic profiling of metagenomic samples are well-studied yet challenging problems. The presence of species belonging to groups without close representation in a reference dataset is particularly challenging. While *k*-mer-based methods have performed well in terms of running time and accuracy, they tend to have reduced accuracy for such novel species. Thus, there is a growing need for methods that combine the scalability of *k*-mers with increased sensitivity.

Results: Here, we show that using locality-sensitive hashing (LSH) can increase the sensitivity of the *k*-mer-based search. Our method, which combines LSH with several heuristics techniques including soft lowest common ancestor labeling and voting, is more accurate than alternatives in both taxonomic classification of individual reads and abundance profiling.

Availability and implementation: CONSULT-II is implemented in C++, and the software, together with reference libraries, is publicly available on GitHub https://github.com/bo1929/CONSULT-II.

1 Introduction

Metagenomic sequencing of microbial communities produces short DNA reads from unknown microorganisms (Handelsman 2004), leading to a need for taxonomic identification based on reference datasets. One approach is to taxonomically identify reads and summarize the results to obtain the taxonomic profile of a sample, showing the relative abundances of taxonomic groups. However, despite the availability of mature read classification and profiling tools, benchmarking has revealed major gaps in the accuracy of existing methods (McIntyre *et al.* 2017, Sczyrba *et al.* 2017, Meyer *et al.* 2019, Ye *et al.* 2019). Precise identification is often hampered by the novelty of queries versus the genome-wide reference datasets and ambiguous matches. In addition, searching against large numbers of genomes is computationally demanding.

Taxonomic identification methods employ various strategies, including *k*-mer matching (Ames *et al.* 2013, Ounit *et al.* 2015, Lu *et al.* 2017, Lau *et al.* 2019, Wood *et al.* 2019), read mapping (Zhu *et al.* 2022), marker-based alignment (Liu *et al.* 2011, Segata *et al.* 2012, Sunagawa *et al.* 2013, Milanese *et al.* 2019), and phylogenetic placement (Truong *et al.* 2015, Asnicar *et al.* 2020, Shah *et al.* 2021). Regardless, they all essentially search for matches between reads in the sample and a reference set. The challenge is that a significant portion of the earth's microbial diversity lacks close representatives in reference datasets (Choi *et al.* 2017), especially in poorly known microbial habitats like seawater or soil (Pachiadaki *et al.* 2019). Thus, most methods use some strategy to seek inexact matches between the query and references and use the results for classification and profiling.

Classification methods often exhibit reduced accuracy for *novel* sequences, which lack representation in reference sets (Nasko *et al.* 2018, von Meijenfeldt *et al.* 2019, Pachiadaki *et al.* 2019, Liang *et al.* 2020). For instance, Rachtman *et al.* (2020) found a leading tool, Kraken-II (Wood *et al.* 2019), faced significant degradation in domain-level classification as the genomic distance to the closest reference increased beyond 10%. Analyses of reads from less commonly sampled environments often fail at classification, even at the phylum level (e.g. Pachiadaki *et al.* 2019). To tackle these challenges, efforts to build more dense reference sets are ongoing (Parks *et al.* 2020, Wu *et al.* 2020, McDonald *et al.* 2023), but these databases remain incomplete compared to the estimated 10^{12} microbial species (Locey and Lennon 2016). In addition, computational challenges arise in searching against large reference sets. Thus, we need accurate and scalable methods of identifying novel sequences with respect to distant reference genomes.

As reference sets grow larger, *k*-mer-based methods become more attractive than alignment-based approaches and phylogenetic placement due to their favorable balance between scalability, ease of use, and high accuracy (McIntyre *et al.* 2017, Ye *et al.* 2019). However, *k*-mer-based methods can be sensitive to reference set completeness if they only allow exact matches. The *k*-mer-based methods that rely on the presence/absence of long *k*-mers can accommodate novel sequences by allowing inexact matches. Kraken-II achieves this by masking some positions in a

Received: 18 July 2023; **Revised:** 17 February 2024; **Editorial Decision:** 10 March 2024; **Accepted:** 14 March 2024

k-mer (default: 7 out of 31). Rachtman *et al.* (2021) showed that novel reads (e.g. those with 10%–15% distance to the closest match) can be identified with higher accuracy by making inexact matches a central feature of the search. The resulting method, CONSULT, uses locality-sensitive hashing (LSH) to partition k-mers in the reference set into fixed-size buckets such that for a given k-mer, the reference k-mers with distance up to a certain threshold are in pre-determined buckets with high probability. By allowing inexact k-mer matching, CONSULT increased sensitivity without compromising precision in the contamination removal application (domain-level classification). However, CONSULT did not perform taxonomic identification at lower levels.

This paper adopts CONSULT and its increased k-mer matching sensitivity to the taxonomic classification problem. CONSULT estimates the Hamming distance (HD) between the query k-mer and its closest reference k-mers, a feature that Kraken-II lacks. Using the distances is the essence of our proposed approach, which we call CONSULT-II. To enable taxonomic classification, we need to track the reference genome(s) associated with each reference k-mer, a feature that CONSULT lacks and can require unrealistically large memory if done naively. We propose a probabilistic method to retain a single taxonomic ID per k-mer, making it possible to fit the database in the memory of modern server nodes. The next challenge is producing a single assignment based on potentially conflicting signals of different k-mers; we address this need using a weighted voting scheme that accounts for distances. Finally, we use a two-level normalization scheme for producing abundance profiles of complex samples using the votes directly. We evaluate the resulting method, CONSULT-II, using a large reference dataset in simulation studies, and show improved accuracy.

2 Algorithm

2.1 Background: CONSULT

The core idea of CONSULT is to find low HD matches efficiently using the bit-sampling LSH method (Har-Peled *et al.* 2012). The use of LSH for finding similar DNA sequences is not new (Buhler 2001, Rasheed *et al.* 2013, Berlin *et al.* 2015, Luo *et al.* 2019). For example, Brown and Truszkowski (2013) addressed the related problem of phylogenetic placement using LSH to limit parts of the reference tree searched. The main focus and novelty of CONSULT, compared to existing work, is being able to search against a large number of reference sequences.

CONSULT tackles the following problem: are there any k-mers in a given set of reference k-mers with HD less than some threshold p to a query k-mer? By default, CONSULT uses $k = 32$ and $p = 3$ (these are adjustable) and can feasibly index a large set of reference k-mers (e.g. 2^{33}).

CONSULT employs two main data structures to represent a set of reference k-mers: an array $\mathcal{K}$ that encodes each 32-mer as a 64-bit number, and l-many (default: $l = 2$) fixed-sized hash tables $\mathbf{H}^1, \ldots, \mathbf{H}^l$ with 4-byte pointers to $\mathcal{K}$ (and extra $n+1$ bits when $|\mathcal{K}| > 2^{32+n}$). Each hash table is a simple $2^{2h} \times b$ matrix (default: $h = 15$ and $b = 7$) where each row is indexed by a hash value and the columns store pointers to k-mers in $\mathcal{K}$. For each hash table $\mathbf{H}^i$, we select h random but fixed positions of a 32-mer as its hash index. Thus, k-mer hashes are computed by simply extracting the corresponding bits from the 64-bit encoding of the k-mer, which is specifically designed to make these extractions efficient. For each query k-mer, the l hashes are computed, pointers from all $\leq b \times l$ entries in the $\mathbf{H}^1, \ldots, \mathbf{H}^r$ are followed to corresponding encodings in $\mathcal{K}$, and the HD is explicitly computed for each such encoding. CONSULT returns a match if there exist k-mers with distance $\leq p$. As such, it has no false positive matches but false negatives (not finding a match) are possible. Using LSH, CONSULT limits the number of HD computations to a constant.

In our bit-sampling scheme (Har-Peled *et al.* 2012), two independent k-mers at HD $= d$ have the same hash with probability $(1 - \frac{d}{k})^h$. Hence, given two independent k-mers, the probability that *at least one* of the hash functions is the same for both k-mers is given by

$$\rho(d) = 1 - \left(1 - \left(1 - \tfrac{d}{k}\right)^h\right)^l. \qquad (1)$$

As desired, for $d \leq p$, $\rho(d)$ is close to 1. For $d \gg p$ and for some small enough p (e.g. $p = 3$), it quickly drops to small values for several choices of l and h. Furthermore, since classification is done at the read level, we have $L - k + 1$ chances for a k-mer match ($L =$ read length). While k-mer dependence across a read hampers computing the probability of having at least one LSH match between a read and a database (and independence assumption would be too inaccurate; see Supplementary Fig. S1), we can still compute the expected number of such matches. Assuming the probability of a mismatch between each base pair of a read and a reference species is $\frac{d}{k}$, the expected number of matching k-mers is $(L - k + 1)\rho(d)$, which can be a large value for realistic settings (Fig. 1a). For example, with the default settings of $k = 32, h = 15, l = 2$, for a 150 bp read at 25% distance from the closest reference, we still expect 3.2 k-mer matches and can potentially classify it (assuming that b is large enough to fit all reference k-mers). Had we used $l = 1$ tables, this expected value would have been 1.6, making it likely to miss many such reads. If we assume 3–4 expected matches provide a sufficiently high probability of at least one match, $l = 1$ would suffice for $d \leq 0.21$, while $l = 3$ and $l = 4$ would only increase our tolerance to $d \leq 0.27$ and $d \leq 0.28$. Given the linear increase in memory with l, we choose $l = 2$ as a tradeoff.

2.2 Overview of the CONSULT-II changes versus CONSULT

To enable taxonomic classification, CONSULT needs to be extended to address several challenges. (i) CONSULT was designed for a fixed reference library size. As a result, all the hashing settings (h, l, b) were fixed for a library of roughly 2^{33} 32-mers. To make the method more usable and flexible, it needs to be adjusted to the size of the input library. CONSULT-II uses several heuristics [see Şapcı *et al.* (2023) and Supplementary Section S1] to estimate an efficient parameter configuration, as a function of the number of k-mers in the reference set and probability of matching two k-mers w.r.t. distance $\rho(d)$. This heuristic enables adjusting the needed memory to reference size. (ii) When building the reference library, we need to keep track of which k-mers belong to which set of taxonomic groups. Since keeping a fine-grained map will lead to an explosion of memory, we need heuristics (detailed below) to store *some* taxonomic information but also to keep the memory manageable. (iii) At the query time, we need some way of combining all the inexact matches from all the k-mers of a read to derive a final identification and to

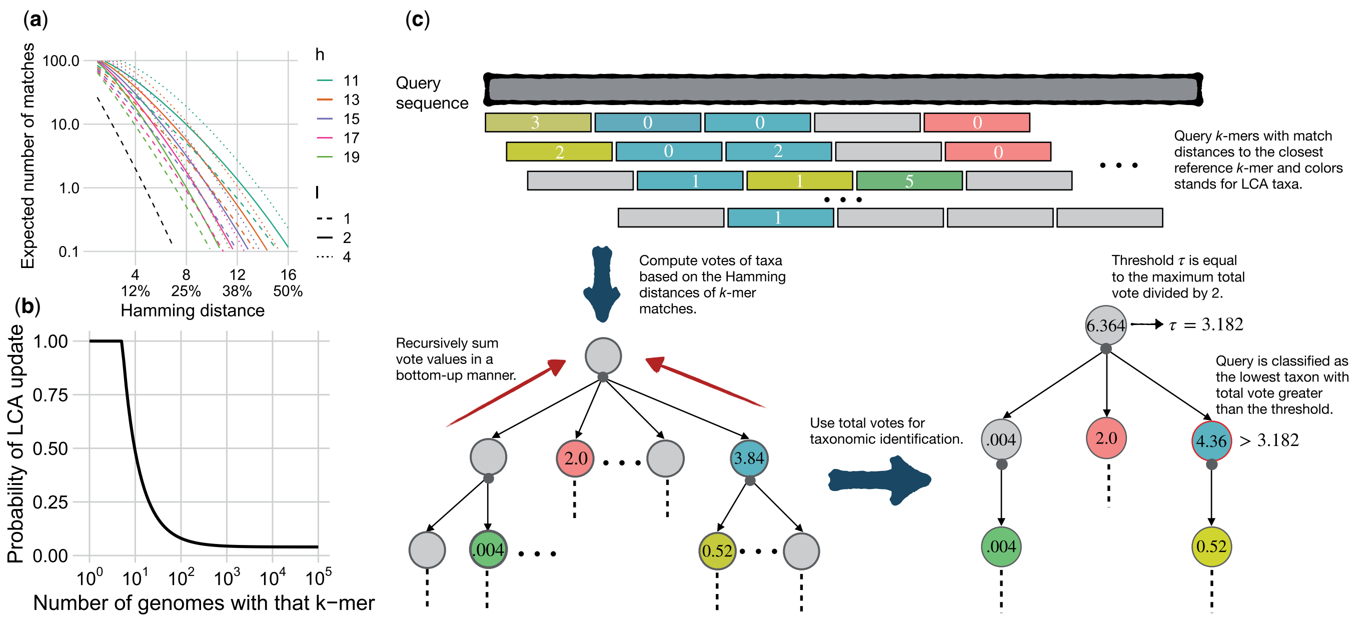
(a)
Expected number of matches
100.0
10.0
1.0
0.1
4 12%
8 25%
12 38%
16 50%
Hamming distance
h
11
13
15
17
19
l
1
2
4
(b)
Probability of LCA update
1.00
0.75
0.50
0.25
0.00
10^0 10^1 10^2 10^3 10^4 10^5
Number of genomes with that k-mer
(c)
Query sequence
3 0 0 0
2 0 2 0
1 1 5
1
Query k-mers with match distances to the closest reference k-mer and colors stands for LCA taxa.
Compute votes of taxa based on the Hamming distances of k-mer matches.
Recursively sum vote values in a bottom-up manner.
2.0
3.84
.004
0.52
Use total votes for taxonomic identification.
Threshold τ is equal to the maximum total vote divided by 2.
6.364 → τ = 3.182
Query is classified as the lowest taxon with total vote greater than the threshold.
.004
2.0
4.36 > 3.182
.004
0.52

Figure 1. (a) The expected number of 32-mers matched by the LSH approach for a short read of length $L=150$ as the normalized distance $\frac{d}{k}$ of the read to the closest match varies: $(L-k+1)\rho(d)$. Lines show different settings of l and h for an infinite b, i.e. all reference k-mers are stored in the library. The black line corresponds to $k=35$, $h=35-7$, and $l=1$, mimicking the default Kraken-II settings. (b) The probability of updating the LCA per each new k-mer goes down with the number of reference genomes in which that k-mer appears; we use $w=4$ and $s=5$ as shown. (c) Overview of the classification algorithm, consisting of three main stages: (i) finding the closest inexact k-mer match using LSH, (ii) computing vote values based on the Hamming distances (HD) and aggregating vote values on the taxonomic tree, and (iii) determining the most specific taxon, i.e. the lowest rank, above the threshold.

summarize the results across all reads in a sample into a final profile.

2.3 Soft lowest common ancestors per k-mer

Considering the density of modern reference datasets, k-mers can appear frequently in several reference species, despite being long (e.g. $k=32$). The required memory for keeping the associated set of taxonomic groups for each k-mer would quickly become infeasible. However, for ≤65 536 taxonomic labels, keeping a single ID requires 2 bytes. Hence, storing one ID per k-mer would consume 16 Gb for our standard libraries with 2^{33} k-mers, which is doable. We choose a single taxonomic ID representing a 'soft' lowest common ancestors (LCA) of all the genomes that include the k-mer using the following procedure.

Let N_i denote the number of genomes that include k-mer $x_i \in \mathcal{K}$, which can be easily computed using pre-processing. At any point in time during the construction, a k-mer $x_i \in \mathcal{K}$ is assigned a 2-byte taxonomic ID, denoted by t_i. We process through each reference genome g; for each k-mer x_i of g, if it can be added to the database, we set or update the taxonomic ID t_i to be the LCA of the current t_i and species of g with probability

$$p_u(N_i) = \min\left\{\frac{w}{\max\{N_i+w-s, w\}} + \frac{1}{s^2}, 1\right\} \tag{2}$$

where w and s parameterize the rate of decrease and the offset of the probability function p_u, respectively. We set $s=5$ and $w=4$ as the default values (see Fig. 1b). Note that the order of processing of the reference genomes has no significance, as every k-mer, including the first encountered, will be ignored with the same probability. Also, k-mers appearing in more than s genomes have a very small, but nonzero, probability of not having a taxonomic ID at all. The goal of the probabilistic soft LCA is to avoid pushing up taxonomic identifications due to errors in the reference libraries, as higher ranks are less informative. Imagine a k-mer that is found exclusively in 20 species of a particular genus, but is also found in one species of a completely different phylum. Using the hard LCA would push the k-mer up to the kingdom level, whereas the soft LCA will stay at the genus rank with 85% probability.

The probability function p_u is a heuristic without a theoretical ground but has two goals. First, it ensures k-mers are assigned an ID if they are *rare* among references (i.e. $N_i \le s = 5$). Second, the probability of ignoring a genome smoothly increases as N_i grows. The $\frac{1}{s^2}$ term is to ensure that each k-mer has a nonzero probability of having a taxonomic ID associated with it, even if it is extremely common.

2.4 Read-level taxonomic identification

For each read, CONSULT-II produces a list of matched k-mers; and for each matched k-mer x, it outputs the soft LCA taxonomic ID and the distances between x and its closest match with the same hash index. To identify a read, we need to derive a single conclusion from all these potentially conflicting signals. We do so by considering each k-mer as providing a *vote* to the corresponding taxonomic ID, but weight votes by the match distance.

Let $\mathcal{T}$ denote the set of all taxonomic IDs, and $\mathcal{K}(t), t \in \mathcal{T}$ be all reference k-mers with t as their soft LCA. Each k-mer x in the set $\mathcal{R}$ of query read k-mers might match multiple k-mers in the reference set $\mathcal{K}$ with varying distances. A match of a lower distance should provide a strong signal. CONSULT-II accounts for this by giving a k-mer $x \in \mathcal{R}$ a vote for the taxonomic ID t using:

$$\mathrm{v}_x(t) = \left(1 - \frac{\min_{y\in\mathcal{K}(t)} hd(x,y)}{k}\right)^k \tag{3}$$

where hd gives the Hamming distance. The voting function (3) drops close to exponentially with distance $\min_{y\in\mathcal{K}(t)} hd(x,y)$. Computing (3) exactly is intractable due to the large size of $\mathcal{K}(t)$. Instead, using LSH, we compare x only to k-mers y with the same hash index as x, finding matches with high probability. Moreover, we let x to vote for only a single taxonomic ID with the minimum distance (breaking ties arbitrarily). As LSH is not effective for high distances, we let a k-mer vote only if its minimum distance is below a threshold $d_{\max}$ (default: $\mathrm{round}(\frac{3p}{2}) = 5$). We set $d_{\max}$ to be higher than p because matches with distance above p might also be found; the LSH guarantees that k-mers with distance $\leq p$ are found with high probability, but more distant k-mers can also be found (see Fig. 1a).

Equation (3), however, is not enough because a vote for a child should also count toward parent ranks. We recursively sum up individual votes in a bottom-up manner using the taxonomic tree to derive a total vote value for each taxonomic ID:

$$\bar{\mathrm{v}}_{\mathcal{R}}(t) = \sum_{x\in\mathcal{R}} \mathrm{v}_x(t) + \sum_{t'\in \mathrm{C}(t)} \bar{\mathrm{v}}_{\mathcal{R}}(t') \tag{4}$$

where $\mathrm{C}(t)$ is the set of children of the taxon t.

By design, the votes $\bar{\mathrm{v}}(t)$ increase for higher ranks and reach their maximum at the root (Fig. 1c). To balance specificity and sensitivity, we require a majority vote. Let $\tau = \frac{1}{2}\max_{t\in\mathcal{T}} \bar{\mathrm{v}}(t)$. CONSULT-II classifies the read with the taxonomic ID $\hat{t}$ that belongs to the lowest rank satisfying the condition $\bar{\mathrm{v}}(\hat{t}) > \tau$. This choice of τ has a special property: only a single taxonomic ID t can exceed τ at a given rank. Therefore, the taxonomic ID predicted by the described classification scheme is unique. Effectively, the classifier starts seeking a taxon at the lowest rank possible but also requires a certain level of confidence; hence, it immediately stops considering upper ranks once the vote value is large enough. In addition, to avoid classification based on only high-distance matches, we require $\bar{\mathrm{v}}(\hat{t})$ to be greater than some small threshold, which we explore in our experimental results.

2.5 Taxonomic profiling

To derive taxonomic abundance profiles, instead of using read identifications, we use votes directly. For each taxonomic rank r (e.g. genus), we first normalize the total votes per read per rank, equalizing the contributions of each read to the profile (if it has any matches). For a read $\mathcal{R}_i$, we simply set

$$\mathrm{v}^*_{\mathcal{R}_i}(t) = \frac{\bar{\mathrm{v}}_{\mathcal{R}_i}(t)}{\sum_{t'\in\mathcal{T}_r} \bar{\mathrm{v}}_{\mathcal{R}_i}(t')} \tag{5}$$

where $\mathcal{T}_r$ is the set of all taxa at rank r. Next, we gather normalized total vote values of all n reads $\mathcal{R}_1, \ldots, \mathcal{R}_n$ in a sample, and normalize again to obtain the final profile. Let $\mathrm{p}^r = [p^r_t]_{t\in\mathcal{T}_r}$ denote the relative abundance profile at rank r, summing up to 1. Then, we can set the relative abundance of taxon t to:

$$p^r_t = \frac{\sum_i^n \mathrm{v}^*_{\mathcal{R}_i}(t)}{\sum_{t'\in\mathcal{T}_r} \sum_i^n \mathrm{v}^*_{\mathcal{R}_i}(t')}. \tag{6}$$

Here, p^r_t estimates the ratio of *reads* belonging to the taxon t in a given sample. Often, we are interested in the relative abundances of *cells* belonging to a taxon t (denoted by $\hat{\mathrm{p}}^r = [\hat{p}^r_t]_{t\in\mathcal{T}_r}$), which needs incorporating genome sizes. We simply do so using:

$$\hat{p}^r_t = \frac{p^r_t l_t^{-1}}{\sum_{t'\in\mathcal{T}_r} p^r_t l_{t'}^{-1}}, \tag{7}$$

where l_t is the average genome length of all references in taxon t.

For both p^r_t and $\hat{p}^r_t$, relative abundances sum up to 1. By default, CONSULT-II relaxes this constraint by including an *unclassified* taxon. This is achieved by propagating votes down to an artificial lineage that corresponds to the unclassified group, as each k-mer match to an LCA taxon provides evidence for its children—but it is unclear which. In other words, we augment the taxonomic tree by adding a lineage under each taxon, which continues until the species rank. Then, all votes to any nonspecies taxon are moved along this lineage to an artificial node at species rank. This is equivalent to changing the denominator of Equation (5) with the total vote at the root of the taxonomic tree.

3 Experimental setup

To benchmark CONSULT-II, we constructed reference libraries using the WoL microbial genomic dataset of Zhu *et al.* (2019a), which is composed of 10 575 species and a reference phylogeny. Five genomes with IDs missing from NCBI were excluded. All methods were run with the same reference set. The hashing parameters of CONSULT-II were set to $h = 15$, $b = 7$, $l = 2$, and $k = 32$ (minimized from canonical 35-mers). For other parameters, default values were used: $w = 4, s = 5$ for LCA probability function p_u (Fig. 1b) and $d_{\max} = 5$ for the vote function. We used default settings for Kraken-II, without masking low-complexity sequences, as Rachtman *et al.* (2020) found default settings to be preferable for query identification. We also constructed the CLARK database using the standard parameters, e.g. $k = 31$, default classification mode, species rank for classification. Note that following Rachtman *et al.* (2021), 100 archaeal genomes were left out from the reference and used as *part of* the query set.

3.1 Experiment 1: controlled novelty

We compared the classification performance of CONSULT-II with two popular methods: Kraken-II (Wood *et al.* 2019) and CLARK (Ounit *et al.* 2015), which are among the leading metagenomic identification tools based on benchmarking studies (McIntyre *et al.* 2017, Sczyrba *et al.* 2017, Meyer *et al.* 2019, Ye *et al.* 2019). Kraken-II maps each k-mer in a read to the LCA of all genomes that contain that k-mer and then counts the mapped k-mers on the taxonomic tree to infer a taxon prediction. CLARK is a supervised sequence classification method that again relies on exact k-mer matching. It uses the notion of discriminative k-mers to build a library of reference genomes. Here, we evaluate accuracy one read at a time, each simulated from a query genome.

Let the *novelty* of a query genome be defined as its minimum genomic nucleotide distance (i.e. one minus average nucleotide identity), as approximated by Mash (Ondov *et al.* 2016), to any genome in the reference database. We refer to this quantity as MinGND. We carefully selected query genomes to span a range of novelty (Supplementary Fig. S2), expecting that more novel queries will be more challenging. We created two sets of queries: bacterial and archaeal. For the bacterial set, we selected 120 bacterial genomes among genomes added to RefSeq after WoL was constructed. Queries range from near-identical to reference genomes to very novel (e.g. 22 with MinGND >0.22; Supplementary Fig. S2). Query genomes span 29 phyla, and most queries are from distinct genera (102 genera across 120 queries); only two query genomes belong to the same species. The 100 archaeal queries were chosen by Rachtman *et al.* (2021) from WoL set using a similar approach and were excluded from the reference set. We generated 150-bp synthetic reads using ART (Huang *et al.* 2012) at higher coverage and then subsampled down to 66 667 reads for each query (i.e. 10Mbp per sample).

We evaluated the predictions of each tool with respect to the NCBI taxonomy. For each read, we evaluate it separately at each taxonomic rank *r*. When the reference library had at least one genome matching the query taxon at rank *r*, we called it a *positive*: *TP* if a tool found the correct label, *FP* if it found an incorrect label, and *FN* if it did not classify at rank *r*. When the reference library did not have any genomes from the query taxon at rank *r*, we called it a *negative*: *TN* if a tool did not classify at rank *r*, *FP* if it classified it, which would necessarily be false. We show the precision $\frac{TP}{(TP+FP)}$, the recall $\frac{TP}{(TP+FN)}$, and F1 $\frac{2TP}{(2TP+FP+FN)}$ which combines both sensitivity and specificity. We ignored queries at levels where the true taxonomic ID given by NCBI is 0, which indicates a missing rank.

3.2 Experiment 2: abundance profiling

We also evaluated the ability of CONSULT-II to perform taxonomic abundance profiling using CAMI-I (Sczyrba *et al.* 2017) and CAMI-II (Meyer *et al.* 2022) benchmarking challenges. We compared tools using metrics provided by the open-community profiling assessment tool (OPAL) (Meyer *et al.* 2019). For CONSULT-II, we allowed unclassified taxa in the profile.

CAMI-I dataset contains five different high-complexity samples, each of size 75 Gbp, which are simulated to mimic the abundance distribution of the underlying microbial communities. Among many metrics, we chose two metrics singled out in the original OPAL paper: the Bray–Curtis dissimilarity between the estimated profile and the true profile and Shannon's equitability as a measure of alpha diversity. We report the summary of these two metrics across five samples. We use CAMI-I dataset for empirical evaluation of our method's heuristics. Here, we used the same reference libraries constructed for controlled novelty experiments from the WoL dataset. As a result, we include only Bracken (Lu *et al.* 2017) and CLARK as alternatives. Bracken extends Kraken-II by combining its taxonomic identification results with Bayesian priors to obtain profiles. For both CLARK and Bracken, we estimated abundance profiles with their default parameters.

On CAMI-II queries, we evaluated CONSULT-II against a host of methods studied by CAMI-II. In particular, we focused on the ten-sample (5 Gbp each) marine dataset and the 100-sample (2 Gbp each) strain-madness dataset. For alternative methods, submitted results were available from CAMI-II. To make comparisons fair, a new CONSULT-II library was constructed using the reference genomes provided under the scope of the challenge (NCBI RefSeq snapshot as of 2019/01/08); among these, we randomly selected $\lfloor\sqrt{n_s}\rfloor$ genomes per each species *s* represented by n_s genome (for a total of 18 381 genomes) and included these in the library. We followed the same evaluation approach as the original paper and compared tools by measuring purity versus completeness and L1 norm versus weighted UniFrac error (Lozupone and Knight 2005). Note that we only included CAMI-II versions of the tools considered, and omitted earlier versions.

4 Results

4.1 Empirical evaluation of CONSULT-II heuristics

4.1.1 Accuracy of *k*-mer matches

We start by evaluating LCAs of matched *k*-mers across different ranks and HD values (Fig. 2a), keeping in mind that incorrect *k*-mer-level matches do not immediately lead to read-level errors. For the queries with low novelty (MinGND <0.06), $\approx 9\%$ of query *k*-mers exactly match the reference and have the correct species-level ID; far fewer exact matches have a soft LCA label at higher taxonomic levels, and the proportion of true matches decreases rapidly with higher HD at all ranks. In contrast, less than 0.5% of the *k*-mers of novel queries (MinGND ≥ 0.22) match any reference, and correct matches peak at HD $=1$ or HD $=2$. For mildly novel genomes ($0.06 \leq$ MinGND <0.22), nonexact matches provide a substantial portion of all correct matches. Unlike true matches, the proportion of false matches tends to increase or remain flat as the HD increases. Nevertheless, in many cases especially at higher taxonomic levels, there are more correct inexact matches with HD ≥ 1 than incorrect ones. For example, 80% of phylum-level inexact matches in the middle novelty range are correct.

4.1.2 Advantages of the soft LCA approach

We next compare the use of hard and soft LCA (Fig. 2b). Overall, soft LCA provides a dramatic improvement compared to naively computing LCA. The improvements are especially large for less novel queries (<0.12 MinGND). Interestingly, at the species rank, hard LCA completely fails while soft LCA has acceptable accuracy for the less novel queries. While a soft LCA approach is clearly helpful, the choice of the ideal probability function is unclear. Since tuning parameters *w* and *s* with a validation set is not practically feasible, we only tested the sensitivity of CONSULT-II performance; settings $w=2$ and $w=4$ are almost identical, with a negligible advantage for $w=4$ (no more than 0.001 in terms of F1). Hence, we use $w=4$ and set it as the default value.

4.1.3 Impact of the total vote on classification

Our majority vote mechanism classifies a read at some level, even if the total vote in Equation (4) is small. However, the total vote correlates with whether a classification is correct (Fig. 2c). In particular, about 35% of the *FP* predictions have $\bar{v}(t)<0.01$ (corresponding to up to two *k*-mers with HD $=5$), compared to $\approx 2\%$ of *TP* predictions. Similarly, around 55% of FPs have $\bar{v}(t)<0.03$ (i.e. less than two matches of HD $=4$), compared to $\approx 6\%$ of TPs. Thus, filtering reads with low $\bar{v}(t)$ reduces the FPs at the expense of

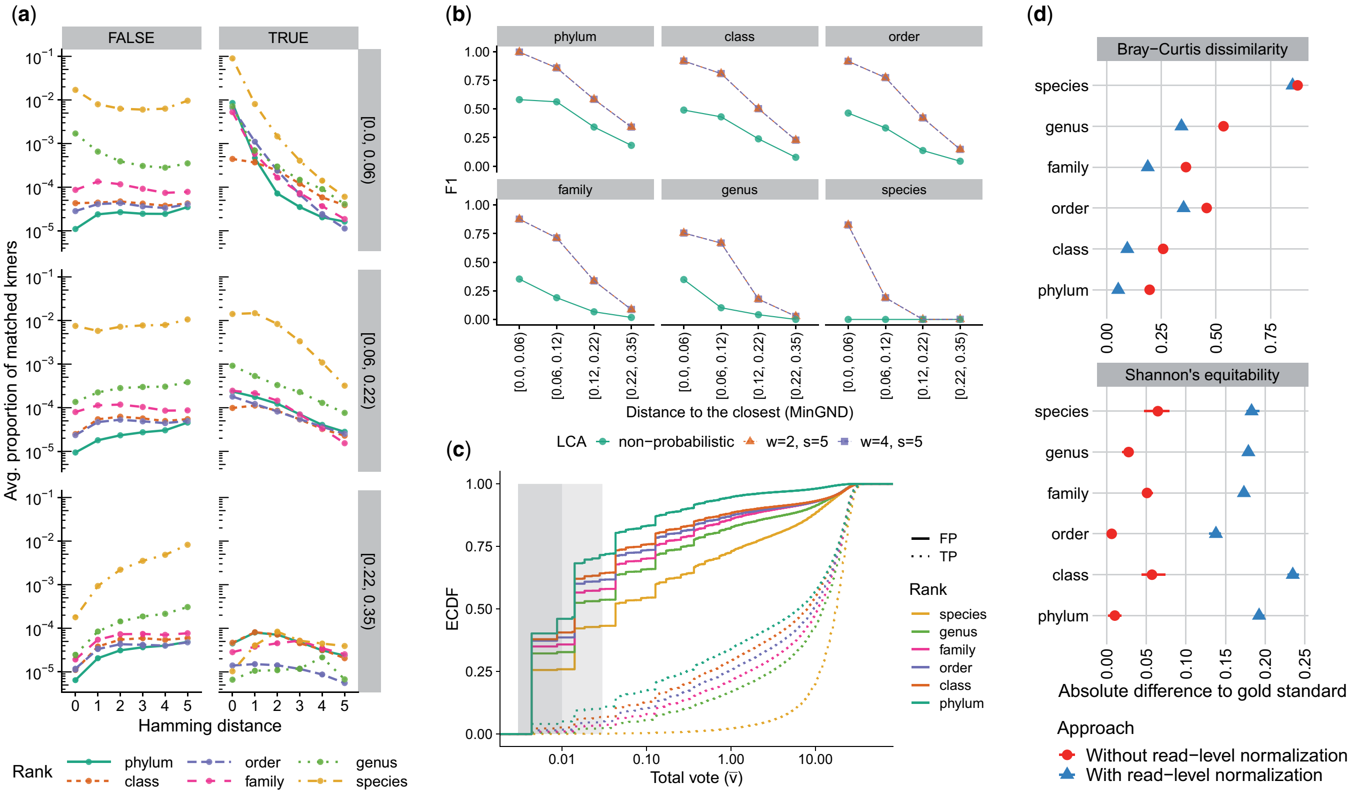

Figure 2. Empirical analysis of LCA update probability p_u and total vote values $\bar{v}(t)$ across different ranks. (a) The proportion of matched *k*-mers across different ranks, novelty bins (box rows, labeled by MinGND range), and Hamming distances. A match is TRUE if the LCA taxon of the matched *k*-mer is from the correct lineage, and matches are only counted for the rank of the LCA taxon. Note that a FALSE *k*-mer match at a particular rank can be correct at higher ranks but is not counted there. The *y*-axis is the ratio between *k*-mer matches of each type and all 119 × 66 667 *k*-mers of a query, averaged over all query genomes. (b) Comparison of soft and hard LCA, based on F1 across different ranks and bins. We compare two parameter settings for the p_u function of soft LCA. (c) The empirical cumulative distribution function (ECDF) of total vote values for correct (*TP*) and incorrect (*FP*) classifications, revealing the predictive power of votes. The shaded areas note 0.01 and 0.03 thresholds, used to filter out spurious classifications. (d) Comparison of two different modes of profiling according to Bray–Curtis dissimilarity and Shannon's equitability. The two measures are computed independently for CONSULT-II's report for each taxonomic rank. Although the default mode [with read-level normalization, using Equation (6)] predicts the actual composition more accurately, the optional mode (without read-level normalization) is a better estimator of Shannon's equitability.

removing some TPs, improving precision and reducing recall (Supplementary Fig. S3b) with only small changes in F1 scores (Supplementary Fig. S3a). To ensure our precision levels are similar to other methods, we avoid classifying reads with low total votes (default: $\bar{v}(t)<0.03$).

4.1.4 Impact of normalization and unclassified on profiling

Since the total vote values at the root vary widely across reads, we face a question in profiling samples: should each read contribute equally to the abundance profile of a given sample? If so, we should normalize total vote values per read, as in Equation (5). Alternatively, we may skip this step and use the total vote values directly in Equation (6). In CONSULT-II `v0.1.1` described in Şapcı *et al.* (2023), we followed the latter approach and skipped Equation (5); we also used a slightly different equation for Equation (6) and took the square root of $\bar{v}(t)$ (Supplementary Section S1.2). Empirically, adding read-level normalization results in a dramatic improvement in accuracy (Fig. 2d) measured by the Bray–Curtis dissimilarity (up to ≈20%). Surprisingly, skipping the read normalization provides extremely accurate estimates of Shannon's equitability. Thus, we keep the nonnormalized version as a nondefault option (since `v0.3.0`). The use of genome sizes [Equation (6) versus Equation. (7)] also improves the profile accuracy (Supplementary Fig. S9). When we added the *unclassified* option to profiles, as much as 35% at the species rank and as little as 6% at the phylum rank were unclassified (Supplementary Fig. S8a). Adding *unclassified* taxa results in slightly more accurate relative abundances in terms of Bray–Curtis dissimilarity (Supplementary Fig. S8b). Thus, we include unclassified taxa in output profiles by default (since `v0.4.0`). Finally, note that Eq. (6) reports independent profiles for each rank. Şapcı *et al.* (2023) instead reported metrics computed at the species level and aggregated to higher ranks (Supplementary Fig. S7).

4.2 Comparison to other methods

4.2.1 Controlled novelty experiments

On the bacterial query set, CONSULT-II has better F1 scores than CLARK and Kraken-II on all levels above species (Fig. 3a). Only for queries with almost identical genomes in the reference set do all methods have high accuracy with a slight advantage at upper ranks for CONSULT-II. As queries become more novel, accuracy drops across all ranks for all methods. However, CONSULT-II degrades much slower and shows clear improvements for novel genomes. For queries with MinGND >0.05, CONSULT-II outperforms Kraken-II and CLARK across all levels above species with substantial margins. Moreover, the improvements become more substantial at higher taxonomic levels. For instance, with MinGND >0.15, CONSULT-II has mean F1 scores that are 0.12, 0.13, 0.14, and 0.19 more than the second-best method (Kraken-II) respectively for family, order, class, and phylum ranks. Similar patterns are observed

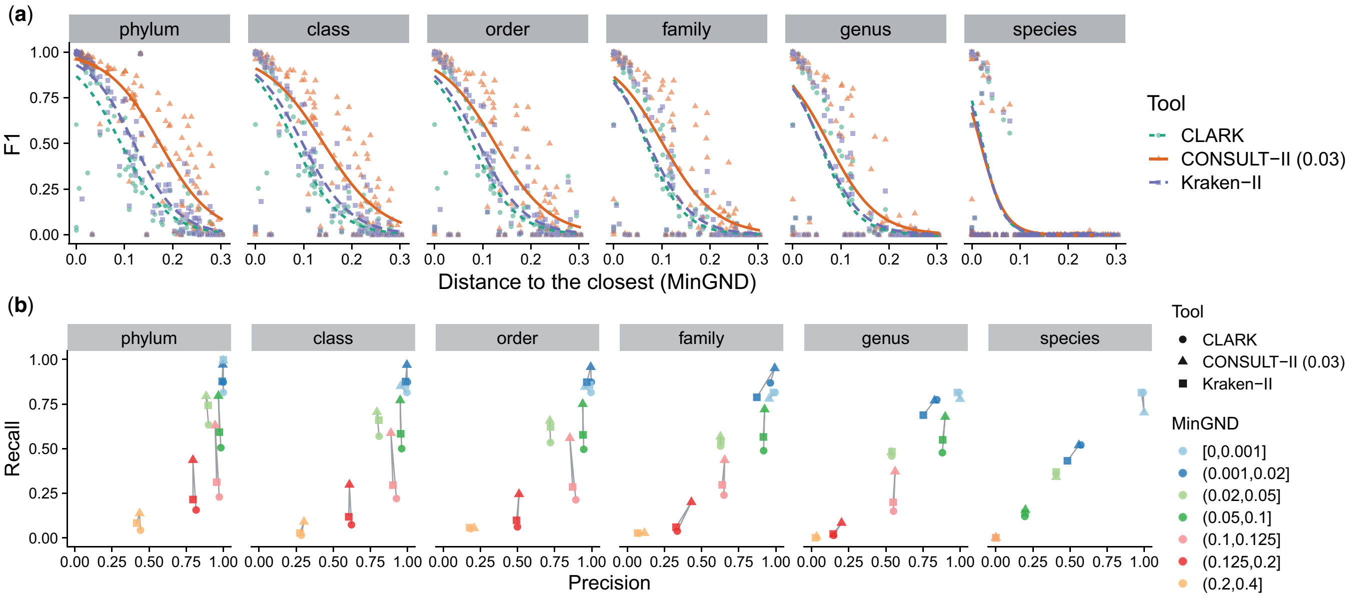

Figure 3. Comparison of CONSULT-II with Kraken-II and CLARK on the controlled novelty experiment with 120 bacterial query genomes and 66 667 reads per genome. (a) Accuracy measured using F1 ($\frac{2TP}{(2TP+FP+FN)}$), showing one dot per query and fitting splines using a generalized linear model. (b) Recall ($\frac{TP}{(TP+FN)}$) versus precision ($\frac{TP}{(TP+FP)}$). Query genomes are binned into seven groups based on their Mash-estimated MinGND. Lines connect results for the same bin for ease of comparison.

when CONSULT-II is run with different total vote thresholds (Supplementary Fig. S3a). Between CLARK and Kraken-II, Kraken-II has a slight advantage in all levels, except at the species level.

The advantage of CONSULT-II is due to higher recall and not precision (Fig. 3b) as the precision levels of methods are comparable in most cases, with only slight differences. At the species level, all methods fail to classify moderately novel queries (MinGND >0.1), and CONSULT-II does not show consistent improvements. At the higher levels, the advantage of CONSULT-II is most clear for novel queries which have the closest reference genome within the $(0.05, 0.2)$ MinGND range. For these queries, CONSULT-II often has equal precision to other methods but much higher recall. For the most novel queries (≥0.2 MinGND) while CONSULT-II shows some improvement, its precision and recall are still not high. Note that the improved recall of CONSULT-II further increases if we decrease the total vote threshold at the expense of precision (see Supplementary Fig. S3b).

Results on the archaeal queries are similar to bacterial queries, with some notable differences (Supplementary Fig. S4). Compared to bacterial queries, all methods tend to have higher F1 scores for queries in the $[0.05, 0.2)$ MinGND range. For all methods, beyond the species rank, the precision tends to be higher regardless of the novelty and recall tends to be lower compared to bacterial queries, especially for less novel genomes. Here, for the bin with the most novel genomes, CONSULT-II has noticeably higher recall and lower precision than alternatives; the gain in the recall offsets the loss in the precision judging by F1.

4.2.2 Profiling results for CAMI-I and CAMI-II challenges

For the CAMI-I challenge, CONSULT-II abundance profiles are consistently better than Bracken and CLARK in terms of the Bray–Curtis score (Fig. 4a). At the species level, all methods have high errors and are comparable. CLARK and Braken are similar across ranks, and the advantage of CONSULT-II is most pronounced at higher levels; the second-best method's error is 40%, 18%, 44%, and 44% higher compared to CONSULT-II at the family to phylum levels, respectively. In terms of Shannon's equitability, which is a measure of the variety and distribution of taxa present in a sample, all methods underestimate it. CONSULT-II and Bracken are comparable, and they both outperform CLARK substantially. Bracken is closer to the gold standard at the phylum level, while CONSULT-II is better at family and below (Fig. 4a).

On the CAMI-II datasets, we compare across 12 methods (Fig. 4b and Supplementary Fig. S6). CONSULT-II is ranked as the second-best performing method for *both* marine and strain-madness datasets, according to rank-invariant weighted UniFrac error, losing to MetaPhlAn 2.9.22 in the strain-madness dataset and to mOTUs 2.5.1 in the marine dataset (Fig. 4b). Note that MetaPhlAn is ranked third on the marine dataset and mOTUs is ranked 9th in the strain-madness dataset (51% higher weighted UniFrac error than CONSULT-II). CONSULT-II is among the top three tools according to L1 norm error in most cases, except at the species rank (Supplementary Fig. S6). For species, CONSULT-II did not have high purity and was not among the best for the L1 norm error, especially in the strain-madness dataset. On the marine dataset, CONSULT-II followed mOTUs, with 0.20 versus 0.13 L1 norm error on average across all ranks. On the strain-madness dataset, it ranked 3rd across all ranks after MetaPhyler and MetaPhlAn with 0.16 versus 0.12 and 0.06 average L1 norm errors across ranks above species (as MetaPhyler lacks species-level profile), respectively.

4.2.3 Resource usages of tools

We benchmark resource usage of all tools over queries selected from 30 genomes generated by simulating 66 667 short reads from each. CONSULT-II and Kraken-II are more than 4× and 9× faster than CLARK, respectively (Fig. 5). While Kraken-II is considerably faster than CONSULT-II (91 versus 390 s), the difference is partly because CONSULT-II splits the task into two independent stages for the sake of backward compatibility with CONSULT. One subprogram finds *k*-mer

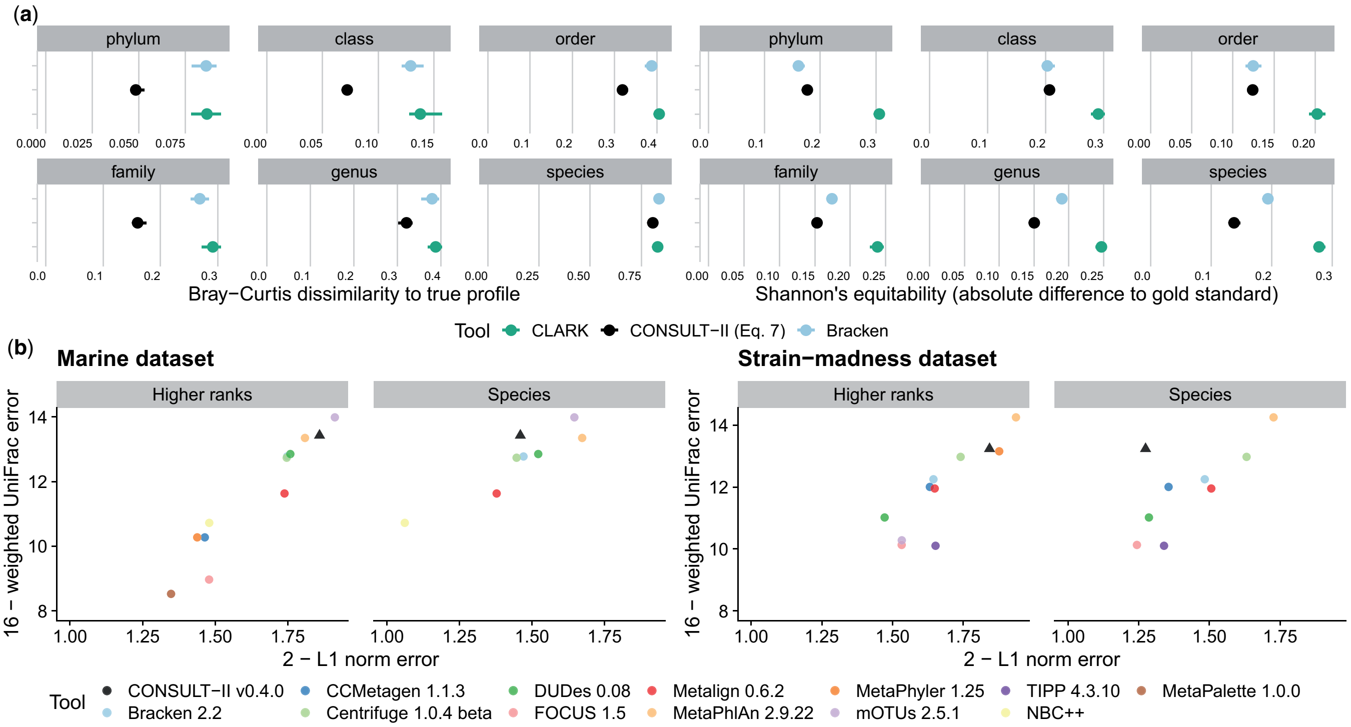

Figure 4. (a) Results on abundance profiling experiments on all five samples of the high-complexity CAMI-I dataset, comparing CONSULT-II with Bracken and CLARK across different taxonomic ranks, using custom libraries constructed with WoL reference database. Left: Bray–Curtis dissimilarity measures the similarity between estimated and true profiles. The smaller the value, the more similar the distributions of abundance profiles are. Right: Shannon's equitability as a measure of alpha diversity. The closer Shannon's equitability of the predicted profile to the gold standard, the better it reflects the actual alpha diversity in the gold standard in terms of the evenness of the taxa abundances. The points are mean values across five samples, and the horizontal bars are for minimum and maximum errors. (b) Performance comparison on CAMI-II benchmarking challenge, marine and strain-madness datasets. Following the original paper, we show the upper bound of L1 norm (2) minus actual L1 norm versus the upper bound of weighted UniFrac error (16) minus actual weighted UniFrac error for genus and phylum ranks. Each data point is the average value over 10 marine samples and 100 strain-madness samples. Higher ranks are mean overall ranks other than species. Metrics are computed using OPAL with default settings and −n. For clarity, axes are cut; see Supplementary Fig. S6 for full results, including all ranks.

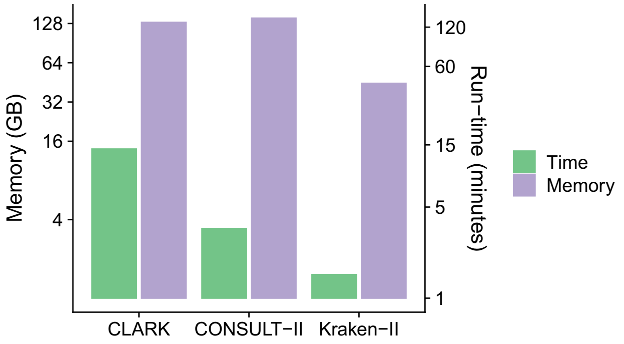

Figure 5. Running time and peak memory usage for 30 queries, each with 66 667 150-bp reads. Computations used 24 threads, performed on a machine with 2.2-GHz AMD EPYC 7742 processors.

matches and writes them to the disk; a second subprogram reads results and performs prediction.

We also observed significant differences in terms of library construction times. Kraken-II was again the fastest tool (≈3.5 h with 96 threads), followed by CLARK (≈10 h with 96 threads). CONSULT-II took ≈19 h (with 128 threads) to construct, and this time was dominated by the soft LCA computation step. Note, however, that library construction is a one-time operation; once constructed, libraries can be distributed.

CONSULT-II, in its default mode, has the highest memory footprint (140 Gb), followed by CLARK (130 Gb). Kraken-II has much better memory efficiency, using only 44 Gb. The CLARK's memory requirement increased substantially during the library building, exceeding 350 Gb. Although it is not as dramatic as CLARK, both CONSULT-II and Kraken-II consumed more memory during the library building (16 and 1 Gb, respectively).

Since CONSULT-II uses more than three times as much memory as Kraken-II, we asked if its improved performance was because of its higher memory usage. To answer this question, we analyzed the performance of CONSULT-II with a much smaller library (32 Gb) by setting $h = 14$ and $b = 10$ to store fewer k-mers (see Supplementary Fig. S5). Compared to the default CONSULT library, the lightweight CONSULT-II had 13%, 12%, 5%, 11%, 10%, and 4% decrease in F1, for phylum to species ranks. Nevertheless, compared to Kraken-II, CONSULT-II achieved 16%, 14%, 10%, 11%, 5% higher F1 scores for phylum to genus ranks, and 6% less at species, despite using less memory (32 versus 44 Gb). Thus, the advantages of CONSULT-II over Kraken-II persist, especially for higher ranks, but at a lower level when using similar memory.

5 Discussion

We introduced CONSULT-II for taxonomic classification and abundance profiling. Our approach uses LSH to efficiently find inexact k-mer matches and the match distances between reference genomes and a query. Heuristics are then used to translate *k*-mer-level matches and their HD to read-level classification and sample-level profiling. While they lack theoretical guarantees, these heuristics performed remarkably well in our experiments and outperformed popular *k*-mer-based methods. In particular, our equations for LCA update probability (2) and vote-versus-distance (3) are based on intuitive assumptions and expectations, without much fine-tuning and very few parameters. Future research should explore alternative approaches, including using machine learning to automatically train parameter-rich functions instead of heuristics. To ensure robustness, it would be essential to evaluate such fine-tuned methods on more varied datasets.

An alternative direction of future work is developing a theoretical framework for translating *k*-mer distances to taxonomic classifications. Connecting taxonomic profiling to distance-based phylogenetic placement could provide a framework to tackle this goal. Such a framework may allow us to go beyond taxonomic identification and could provide alignment-free phylogenetic placement of reads, as others have attempted (Blanke and Morgenstern 2020). Finally, future work should address the high memory consumption of CONSULT-II compared to alternatives, perhaps by smart subsampling of *k*-mers.

Supplementary data

Supplementary data are available at *Bioinformatics* online.

Conflict of interest

None declared.

Funding

The funding for this work was provided by the National Institute of Health (NIH) grant number R35GM142725 and by a research grant by Minderoo Foundation to S.M.

Data availability

CONSULT-II is implemented in C++, the source code is available at https://github.com/bo1929/CONSULT-II. All results and auxiliary data together with analysis scripts used to create figures are available at https://github.com/bo1929/shared.CONSULT-II. Datasets were derived from Sczyrba *et al.* (2017), Meyer *et al.* (2022) and Zhu *et al.* (2019a).

References

Ames SK, Hysom DA, Gardner SN *et al.* Scalable metagenomic taxonomy classification using a reference genome database. *Bioinformatics*2013;**29**:2253–60.

Asnicar F, Thomas AM, Beghini F *et al.* Precise phylogenetic analysis of microbial isolates and genomes from metagenomes using PhyloPhlAn 3.0. *Nat Commun* 2020;**11**:2500.

Berlin K, Koren S, Chin C-S *et al.* Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nat Biotechnol* 2015;**33**:623–30.

Blanke M, Morgenstern B. Phylogenetic placement of short reads without sequence alignment. bioRxiv, https://doi.org/10.1101/2020, 2020, preprint: not peer reviewed.

Brown D, Truszkowski J. LSHPlace: fast phylogenetic placement using locality-sensitive hashing. In: *Pacific Symposium on Biocomputing*, Hawaii, USA, 2013, 310–9.

Buhler J. Efficient large-scale sequence comparison by locality-sensitive hashing. *Bioinformatics* 2001;**17**:419–28.

Choi J, Yang F, Stepanauskas R *et al.* Strategies to improve reference databases for soil microbiomes. *ISME J* 2017;**11**:829–34.

Handelsman J. Metagenomics: application of genomics to uncultured microorganisms. *Microbiol Mol Biol Rev* 2004;**68**:669–85.

Har-Peled S, Indyk P, Motwani R *et al.* Approximate nearest neighbors: towards removing the curse of dimensionality. *Theory of Comput* 2012;**8**:321–50.

Huang W, Li L, Myers JR *et al.* ART: a next-generation sequencing read simulator. *Bioinformatics* 2012;**28**:593–4.

Lau A-K, Dörrer S, Leimeister C-A *et al.* Read-SpaM: assembly-free and alignment-free comparison of bacterial genomes with low sequencing coverage. *BMC Bioinformatics* 2019;**20**:638.

Liang Q, Bible PW, Liu Y *et al.* DeepMicrobes: taxonomic classification for metagenomics with deep learning. *NAR Genom Bioinform* 2020;**2**:lqaa009.

Liu B, Gibbons T, Ghodsi M *et al.* MetaPhyler: Taxonomic profiling for metagenomic sequences. In: *2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, Hong Kong, China, 2011, 95–100.

Locey KJ, Lennon JT. Scaling laws predict global microbial diversity. *Proc Natl Acad Sci USA* 2016;**113**:5970–5.

Lozupone C, Knight R. UniFrac: a new phylogenetic method for comparing microbial communities. *Appl Environ Microbiol* 2005;**71**:8228–35.

Lu J, Breitwieser FP, Thielen P *et al.* Bracken: estimating species abundance in metagenomics data. *PeerJ Computer Sci* 2017;**3**:e104.

Luo Y, Yu YW, Zeng J *et al.* Metagenomic binning through low-density hashing. *Bioinformatics* 2019;**35**:219–26.

McDonald D, Jiang Y, Balaban M *et al.* Greengenes2 unifies microbial data in a single reference tree. *Nat Biotechnol* 2023. https://doi.org/10.1038/s41587-023-01845-1.

McIntyre ABR, Ounit R, Afshinnekoo E *et al.* Comprehensive benchmarking and ensemble approaches for metagenomic classifiers. *Genome Biol* 2017;**18**:182.

Meyer F, Bremges A, Belmann P *et al.* Assessing taxonomic metagenome profilers with OPAL. *Genome Biol* 2019;**20**:51.

Meyer F, Fritz A, Deng Z-L *et al.* Critical assessment of metagenome interpretation: the second round of challenges. *Nat Methods* 2022;**19**:429–40.

Milanese A, Mende DR, Paoli L *et al.* Microbial abundance, activity and population genomic profiling with mOTUs2. *Nat Commun* 2019;**10**:1014.

Nasko DJ, Koren S, Phillippy AM *et al.* RefSeq database growth influences the accuracy of k-mer-based lowest common ancestor species identification. *Genome Biol* 2018;**19**:165.

Ondov BD, Treangen TJ, Melsted P *et al.* Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biol* 2016;**17**:132.

Ounit R, Wanamaker S, Close TJ *et al.* CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC Genomics* 2015;**16**:236.

Pachiadaki MG, Brown JM, Brown J *et al.* Charting the complexity of the marine microbiome through single-cell genomics. *Cell* 2019;**179**:1623–35.e11.

Parks DH, Chuvochina M, Chaumeil P-A *et al.* A complete domain-to-species taxonomy for Bacteria and Archaea. *Nat Biotechnol* 2020;**38**:1079–86.

Rachtman E, Balaban M, Bafna V *et al.* The impact of contaminants on the accuracy of genome skimming and the effectiveness of exclusion read filters. *Mol Ecol Resour* 2020;**20**:1755–0998. 13135.

Rachtman E, Bafna V, Mirarab S *et al.* CONSULT: accurate contamination removal using locality-sensitive hashing. *NAR Genom Bioinform* 2021;**3**:lqab071.

Rasheed Z, Rangwala H, Barbará D *et al.* 16S rRNA metagenome clustering and diversity estimation using locality sensitive hashing. *BMC Syst Biol* 2013;**7**:S11.

Şapcı AOB, Rachtman E, Mirarab S. CONSULT-II: taxonomic identification using locality sensitive hashing. In: Jahn K, Vinař T (eds.), *Comparative Genomics*. Cham: Springer Nature Switzerland. 2023, 196–214.

Sczyrba A, Hofmann P, Belmann P *et al.* Critical assessment of metagenome interpretation—a benchmark of metagenomics software. *Nat Methods* 2017;**14**:1063–71.

Segata N, Waldron L, Ballarini A *et al.* Metagenomic microbial community profiling using unique clade-specific marker genes. *Nat Methods* 2012;**9**:811–4.

Shah N, Molloy EK, Pop M *et al.* TIPP2: metagenomic taxonomic profiling using phylogenetic markers. *Bioinformatics* 2021;**37**:1839–45.

Sunagawa S, Mende DR, Zeller G *et al.* Metagenomic species profiling using universal phylogenetic marker genes. *Nat Methods* 2013; **10**:1196–9.

Truong DT, Franzosa EA, Tickle TL *et al.* MetaPhlAn2 for enhanced metagenomic taxonomic profiling. *Nat Methods* 2015;**12**:902–3.

von Meijenfeldt FAB, Arkhipova K, Cambuy DD *et al.* Robust taxonomic classification of uncharted microbial sequences and bins with CAT and BAT. *Genome Biol* 2019;**20**:217.

Wood DE, Lu J, Langmead B *et al.* Improved metagenomic analysis with Kraken 2. *Genome Biol* 2019;**20**:257.

Wu S, Sun C, Li Y *et al.* GMrepo: a database of curated and consistently annotated human gut metagenomes. *Nucleic Acids Res* 2020; **48**:D545–53.

Ye SH, Siddle KJ, Park DJ *et al.* Benchmarking metagenomics tools for taxonomic classification. *Cell* 2019;**178**:779–94.

Zhu Q, Mai U, Pfeiffer W *et al.* Phylogenomics of 10,575 genomes reveals evolutionary proximity between domains Bacteria and Archaea. *Nat Commun* 2019a;**10**:5477.

Zhu Q, Huang S, Gonzalez A *et al.* Phylogeny-aware analysis of metagenome community ecology based on matched reference genomes while bypassing taxonomy. *mSystems* 2022;**7**:e0016722.