Bioinformatics, 2022, 1–8
doi: DOI HERE
Advance Access Publication Date: Day Month Year
Paper

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

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

[1]Bioinformatics and Systems Biology Graduate Program, University of California, San Diego, Gilman Drive, 92093, CA, USA and [2]Electrical and Computer Engineering, University of California, San Diego, Gilman Drive, 92093, CA, USA

*Corresponding author. smirarab@ucsd.edu

FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

Taxonomic classification of metagenomic reads is a well-studied yet challenging problem. Identifying species belonging to ranks without close representation in a reference dataset are in particular challenging. While k-mer-based methods have performed well in terms of running time and accuracy, they have reduced accuracy for novel species. 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 LCA labeling and voting is more accurate than alternatives in both taxonomic classification and profiling.

Key words: Metagenomics, Taxonomic classification, Abundance profiling

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. Identifying reads at the lowest taxonomic level is the goal, and results can be summarized to obtain the taxonomic profile of a sample, showing the relative abundances of taxonomic groups. However, despite the availability of mature classification and profiling tools, benchmarking has revealed major gaps in the accuracy of existing methods (McIntyre *et al.*, 2017; Meyer *et al.*, 2019; Sczyrba *et al.*, 2017; 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; Wood *et al.*, 2019; Lau *et al.*, 2019; Lu *et al.*, 2017), read mapping (Zhu *et al.*, 2022), marker-based alignment (Liu *et al.*, 2011; Milanese *et al.*, 2019; Segata *et al.*, 2012; Sunagawa *et al.*, 2013), and phylogenetic placement (Asnicar *et al.*, 2020; Shah *et al.*, 2021; Truong *et al.*, 2015). 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 (Liang *et al.*, 2020; von Meijenfeldt *et al.*, 2019; Nasko *et al.*, 2018; Pachiadaki *et al.*, 2019). 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 (McDonald *et al.*, 2023; Parks *et al.*, 2020; Wu *et al.*, 2020), but these databases remain incomplete compared to the estimated 10^{12} microbial species (Locey and Lennon, 2016). Additionally, 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 (Ye *et al.*, 2019; McIntyre *et al.*, 2017). 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 k-mer (default: 7 out of 31). Recently, 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

© The Author 2022. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

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 within 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 identification, 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 making a single call based on potentially conflicting signals coming from 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 much-improved accuracy.

Algorithm

Background: CONSULT

The core idea of CONSULT is to find low HD matches efficiently using LSH. The use of LSH for finding similar DNA sequences is not new (Berlin *et al.*, 2015; Buhler, 2001; Rasheed *et al.*, 2013; 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-mer with Hamming distances (HD) less than some threshold p to a query k-mer? By default, CONSULT uses k=32 and p=3, but parameters can change. CONSULT 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 $\mathbf{H}^1, \ldots, \mathbf{H}^l$ 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 positives matches but false negatives (not finding a match) are possible. By employing the LSH scheme, CONSULT limits the number of HD computations to a constant.

Using this simple LSH scheme, two k-mers at HD $=d$ have the same hash with probability $(1 - {}^{d}/_{k})^h$. Hence, given two k-mers, the probability that *at least one* of the hash functions is the same for both k-mers is given by

$$\rho(d) = 1 - (1 - (1 - \frac{d}{k})^h)^l. \quad (1)$$

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 do not need every k-mer of the read to match to be able to classify a read. Each read includes $L - k + 1$ k-mers where L is the length of the read. Therefore, when the probability of a mismatch between a read and a reference species is ${}^{d}/_{k}$, the expected number of matching k-mers is $(L-k+1)\rho(d)$, which can be a large value for realistic settings (Fig. 1a). Note that, with the default settings of k=32, h=15, l=2, for a short read at distance 25% distance from the closest reference, we still expect 3.2 k-mer matches and can potentially classify it.

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

Soft lowest common ancestors (LCA) 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 $\leq$ 65536 taxonomic labels, keeping a single ID requires 2 bytes. Hence, storing one ID per k-mer would consume 16Gb for our standard libraries with 2^{33} k-mers, which is doable. We choose a single taxonomic ID representing a "soft" 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 r; for each k-mer x_i of r, if it can be added to

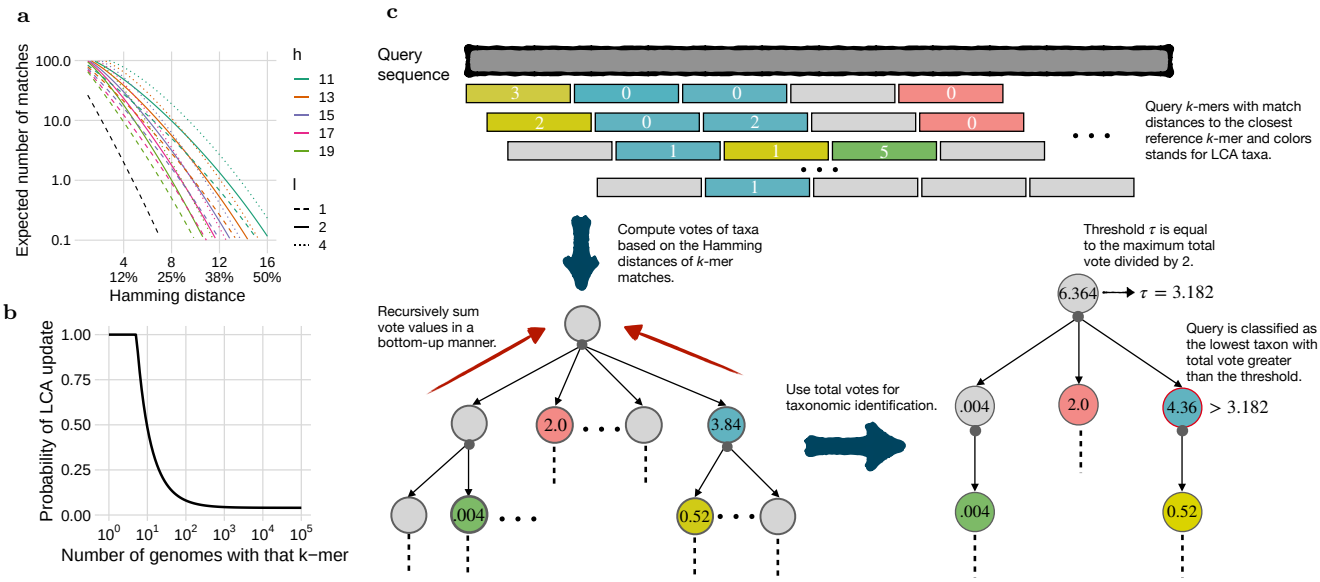

Fig. 1: (a) The expected number of 32-mers matched by the LSH approach for a short read of length L=150 as the normalized distance $^d/_k$ of the read to the closest match varies: $(L-k+1)\rho(d)$. Lines show different settings of l and h for an infinite b, i.e., all reference k-mers are stored in the 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 iii) determining the most specific taxon, i.e., lowest rank, above the threshold.

the database, we set or update the taxonomic ID t_i to be the LCA of the current t_i and species r 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 non-zero, 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 due to incorrect assembly (or HGT), 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., $N_i \leq s$=5). Second, the probability of ignoring a genome smoothly increases as N_i grows. The $^1/_{s^2}$ term is to ensure that each k-mer has a non-zero probability of having a taxonomic ID associated with it, even if it is extremely common.

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:

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

where hd gives HD. 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: round($^{3p}/_2$) =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 towards 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:

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

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. Additionally, 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.

Taxonomic profiling

To derive taxonomic abundance profiles, instead of using read identifications, we use votes directly. For each taxonomic rank l (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

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

where $\mathcal{T}_l$ is the set of all taxa at rank l. 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 $\mathsf{p}^l = \left[p^l_t\right]_{t\in\mathcal{T}_l}$ denote the abundance profile at rank l summing up to one. Then, CONSULT-II sets the relative abundance of taxon t to:

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

Experimental setup

To benchmark CONSULT-II, we constructed reference libraries using the ToL microbial genomic dataset of Zhu *et al.* (2019), 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.

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 leading metagenomic identification tools based on benchmarking studies (Ye *et al.*, 2019; Meyer *et al.*, 2019; Sczyrba *et al.*, 2017; McIntyre *et al.*, 2017). Kraken-II maps each k-mer in a read to the LCA of all genomes which 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.

We created two sets of queries: bacterial and archaeal. The 100 archaeal queries were chosen by Rachtman *et al.* (2021) and were excluded from the reference set. For the bacterial set, we selected a set of 120 bacterial genomes among genomes added to RefSeq after ToL was constructed. We carefully selected query genomes to span a range of different distances to the closest species in the reference dataset (Fig. S1) as measured by Mash (Ondov *et al.*, 2016). Queries include 13 that are very similar to reference genomes and 23 that are very novel – with distances to the closest greater than 0.22. Query genomes span 29 phyla, and most queries are from distinct genera (102 genera across 120 queries) and only one species contributes two queries. We generated 150bp synthetic reads using ART (Huang *et al.*, 2012) at higher coverage, and then subsampled down to 66667 reads for each query, which corresponds to $\approx 2\times$ coverage for bacterial genomes.

We evaluated the predictions of each tool with respect to the NCBI taxonomy. For each read, we evaluate it separately at each taxonomic level l. When the reference library had at least one genome matching the query taxon at level l, 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 level l. When the reference library did not have any genomes from the query taxon at level l, we called it a *negative*: TN if a tool did not classify at level l, FP if it classified it, which would necessarily be false. We show the precision $TP/(TP+FP)$, the recall $TP/(TP+FN)$, and F1 $2TP/(2TP+FP+FN)$ 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.

Experiment 2: abundance profiling

We also evaluated the ability of CONSULT-II to perform taxonomic abundance profiling by comparing it with Bracken (Lu *et al.*, 2017) and CLARK. Bracken extends Kraken-II by combining its taxonomic identification results with Bayesian priors. For both CLARK and Bracken, we estimated abundance profiles with their default parameters. We used a dataset from the CAMI-I (Sczyrba *et al.*, 2017) benchmarking challenge. This dataset contains five different high-complexity samples, each of size 75Gbp, which are simulated to mimic the abundance distribution of the underlying microbial communities. We compared profiling tools using metrics provided by the open-community profiling assessment tool (OPAL) (Meyer *et al.*, 2019). Among many, we chose two metrics singled out in the original paper: the Bray-Curtis dissimilarity between the estimated profile and the true profile and the Shannon's equitability as a measure of alpha diversity. We report the summary of these two metrics across five samples.

Results

Empirical evaluation of CONSULT-II heuristics

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 ($<$ 6%), around 9% of query

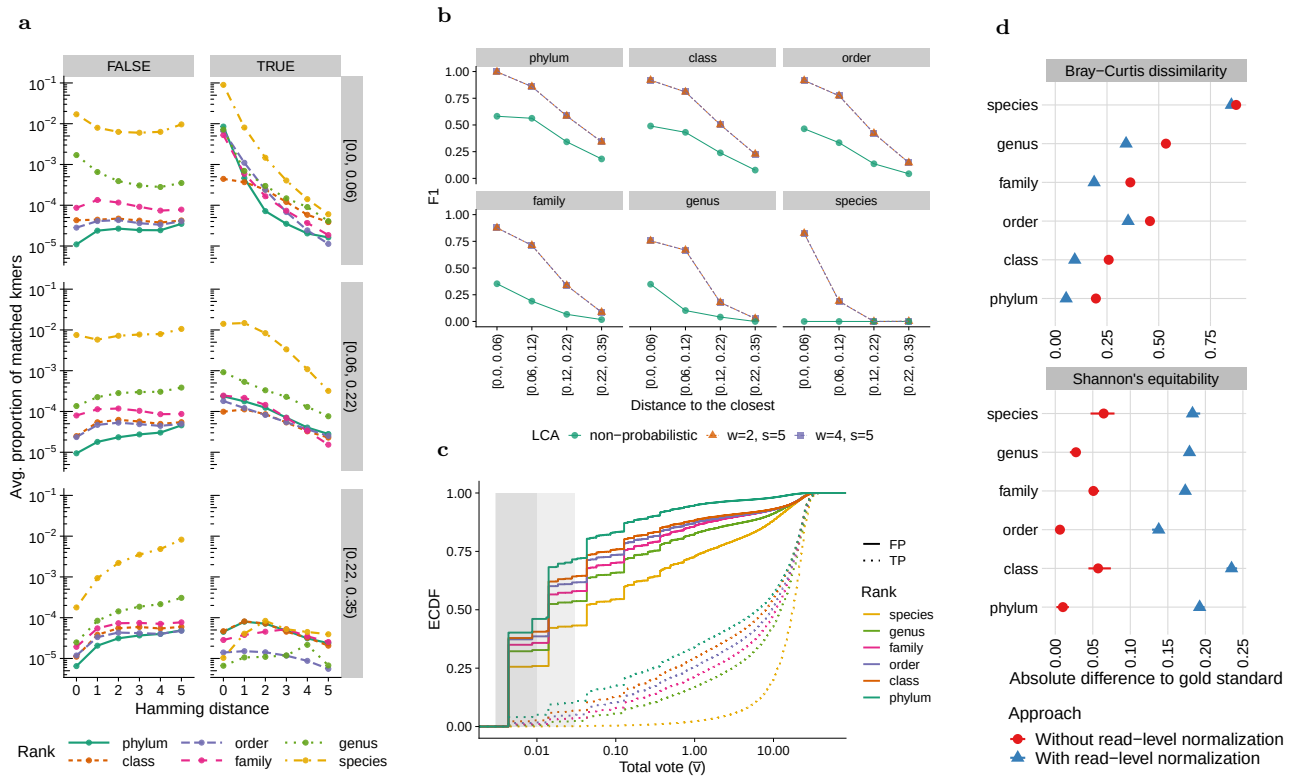

Fig. 2: Empirical analysis of LCA update probability p_u and total vote values $\bar{\mathrm{v}}(\mathrm{t})$ across different ranks. (a) The proportion of matched k-mers across different ranks, novelty bins, and Hamming distances. A match is true if the LCA taxon of the matched k-mer is from the correct 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 119×66667 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) predicts the actual composition more accurately, the optional mode (without read-level normalization) is a better estimator of the Shannon's equitability.

k-mers exactly match the reference and have the correct species-level ID; far fewer exact-matching k-mers 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, for the most novel queries ($>$ 22% distance), less than 0.5% of their k-mers match any reference, and correct matches peak at HD= 1 or HD= 2. For mildly novel genomes in between the two extremes, non-exact 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. For example, for the least novel genomes, 1.7% of query k-mers with an exact match have a wrong soft LCA label compared to $\approx$ 4% for inexact matches with HD $\geq$ 1. Nevertheless, in many cases, there are more correct inexact matches with HD $\geq$ 1 than incorrect ones, especially at higher levels. For example, 80% of phylum-level inexact matches in the middle novelty range are correct.

Advantages of the soft LCA approach

We next compare the use of hard and soft LCA with different parameter configurations (Fig. 2b). Overall, soft LCA provides a dramatic improvement compared to naively computing LCA. The improvements are especially large for less novel queries with distances to closest < 0.12. 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 to the rate of decrease, parameterized by w in p_u. The resulting performances for w=2 and w=4 are almost identical, with a negligible advantage of w=4 (no more than 0.001 in terms of F1). Hence, we use w=4 and set it as the default value.

Impact of the total vote on classification

Our majority vote mechanism will classify a read at some level, even if the total votes in Eq. (4) are very small at the root of the taxonomy. Our empirical examination showed that the total vote correlates with whether a classification is correct (Fig. 2c). In particular, about 35% of the FP predictions have $\bar{\mathrm{v}}(t) < 0.01$ (corresponding to up to two k-mers with HD=5), compared to $\approx$2% of TP predictions. Similarly, around 55% of FPs have $\bar{\mathrm{v}}(t) < 0.03$ (i.e., less than two matches of HD=4), compared to

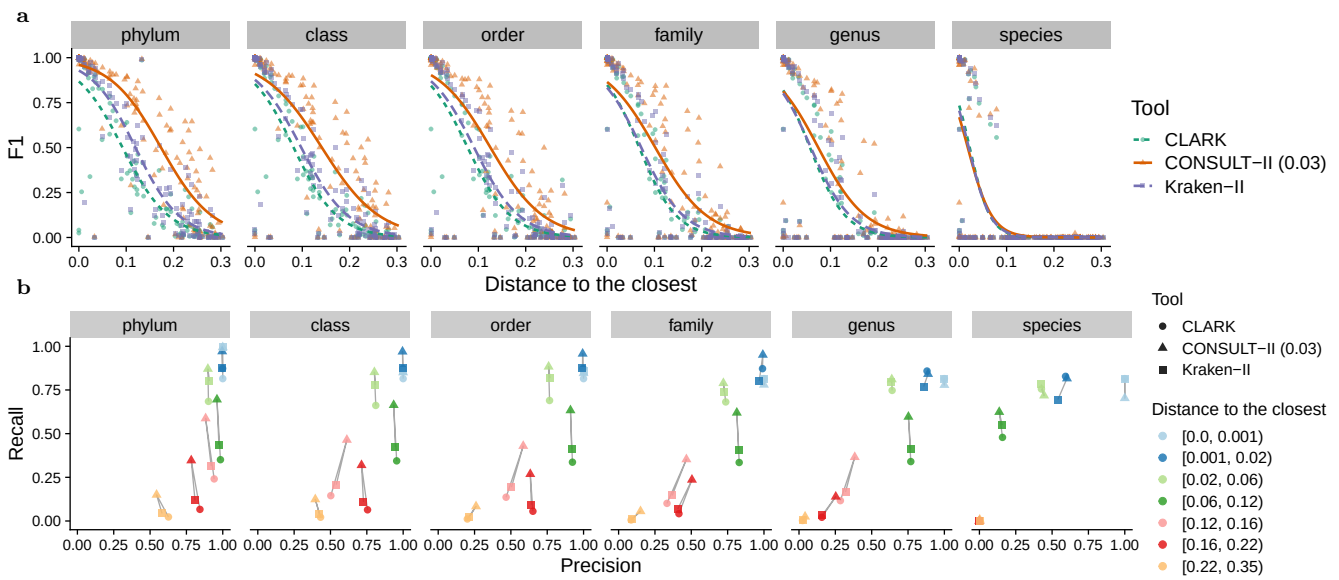

Fig. 3: Comparison of CONSULT-II with Kraken-II and CLARK on the controlled novelty experiment with 120 bacterial query genomes and 66667 reads per genome. (a) Accuracy measured using F1 ($^{2TP}/_{(2TP+FP+FN)}$), showing one dot per query and fitting splines using a generalized linear model. (b) Recall ($^{TP}/_{(TP+FN)}$) versus precision ($^{TP}/_{(TP+FP)}$). Query genomes are binned into 7 groups based on their Mash distance to their closest match in the reference set. Lines connect results for the same bin for ease of comparison.

≈6% of TPs. Thus, filtering reads with low $\bar{\mathrm{v}}(t)$ reduces the FPs at the expense of removing some TPs, improving precision and reducing recall. Comparing different levels of filtering shows very different precision and recall levels (Fig. S2b) but similar F1 scores (Fig. S2a). There is a small but consistent decrease in F1 with more aggressive filtering. Nevertheless, to reduce false positives and to ensure our precision levels are similar to other methods, we avoid classifying reads with low total votes (default: $\bar{\mathrm{v}}(t) < 0.03$).

Impact of normalization 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 Eq. (5). Alternatively, we may skip this step and use the total vote values directly in Eq. (6). In CONSULT-II `v0.1.1` described in Şapcı *et al.* (2023), we followed the latter approach and skipped Eq. (5); we also used a slightly different equation for Eq. (6) and took the square root of total vote values (Section S1.2). Our empirical evaluations reveal that adding read-level normalization, as proposed here, results in a dramatic improvement in profile accuracy (Fig. 2d). The Bray-Curtis dissimilarity can improve up to ≈20%. Surprisingly, skipping the read normalization provides extremely accurate estimates of Shannon's equitability (which does not depend on any notion of the correctness of taxonomic labels). While we do not believe Shannon's equitability should be the main focus of profiling, we nevertheless keep the non-normalized version option as a non-default mode of CONSULT-II (`v0.3.0`). Finally, note that CONSULT-II reports independent profiles for each rank using Eq. (6). In contrast, in Şapcı *et al.* (2023), we reported metrics computed solely on the species-level profile and aggregated to higher levels (see Fig. S4 for comparisons).

Comparison to other methods

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; even there, CONSULT-II has a slight advantage for upper ranks such as phylum and class. 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 distance to the closest reference > 0.05, CONSULT-II starts to outperform Kraken-II and CLARK across all levels above species with substantial margins. Moreover, the improvements become more substantial at higher taxonomic levels. For instance, for genomes with distance to the closest > 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 when CONSULT-II is run with different total vote thresholds (Fig. S2a). Between CLARK and Kraken-II, Kraken-II has a slight advantage in all levels, except at the species level.

Comparing the precision and recall of the methods reveals that the advantage of our tool is due to higher recall and not precision (Fig. 3b). The precision of methods is comparable in most cases, is slightly higher for CONSULT-II in some cases (e.g., family level), and is slightly worse in other cases (e.g., phylum level). At the species level, all methods fail to classify novel queries with distance to the closest ≥ 0.12, and have low precision even for slightly novel genomes. At this level, CONSULT-II does not show any sign of improvement. At the higher levels, the advantage of CONSULT-II is most clear for novel queries with distance to the closest between 0.06 and 0.22. For these queries, CONSULT-II often has equal precision to other methods but much higher recall. For the most

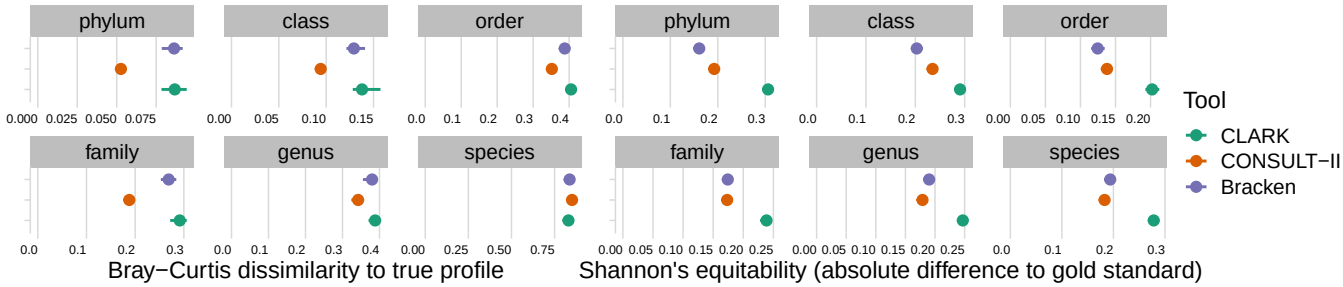

Fig. 4: Results on abundance profiling experiments on all five samples of the high-complexity CAMI-I dataset, comparing CONSULT-II with Bracken and CLARK across different taxonomic ranks. Left: Bray-Curtis dissimilarity measures the similarity between estimated and true profiles. The smaller the value, the more similar distributions of abundance profiles are. Right: Shannon's equitability as a measure of alpha diversity. The closer the 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 standard error.

novel queries with distance to the closest ≥ 0.22, while CONSULT-II shows some improvement, its precision and recall are both too low to make classifications useful. Note that the improved recall of CONSULT-II is despite using a more conservative version of the total-vote filtering; decreasing the total vote threshold can further increase the recall at the expense of precision (see Fig. S2b).

Results on the archaeal queries are similar to bacterial queries, with some notable differences (Fig. S3). Compared to bacterial queries, all methods tend to have higher F1 scores for slightly novel genomes (e.g., distance to the closest in $[0.05, 0.15)$). Also, beyond the species rank, the precision tends to be higher and recall tends to be lower compared to bacterial queries for all methods. 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.

Profiling results for CAMI-I dataset

On the CAMI-I dataset, CONSULT-II abundance profiles are more similar to the true microbial composition than alternative methods are (Fig. 4). Above the species level, it clearly and consistently outperforms Bracken and CLARK in terms of the Bray-Curtis score. At the species level, all methods have high errors and are comparable. CLARK and Kraken-II are similar across ranks, and the advantage of CONSULT-II is most pronounced at higher levels; its error is 30%, 9%, 27%, and 39% lower than the second-best method at the family to phylum levels, respectively.

In terms of the Shannon's equitability, which is a measure of the variety and distribution of taxa present in a sample, all methods underestimate Shannon's equitability. CONSULT-II and Bracken are comparable, and they both outperform CLARK across all ranks, with a significant difference, e.g., as much as 10% at the phylum level. Bracken has closer Shannon's equitability to gold-standard at the phylum, class, and order ranks, and its difference with CONSULT-II varies between 1% and 3% (Fig. 4). At the species and genus levels, CONSULT-II is slightly better than the second-best method, Bracken, by 1%.

Resource usages of tools

Measured over 30 microbial queries, each with 66667 short reads, and using 24 threads, CONSULT-II and Kraken-II are more than $\times 15$ and $\times 5$ faster than CLARK, respectively. While CONSULT-II is faster than Kraken-II, the difference is mostly due to Kraken-II's

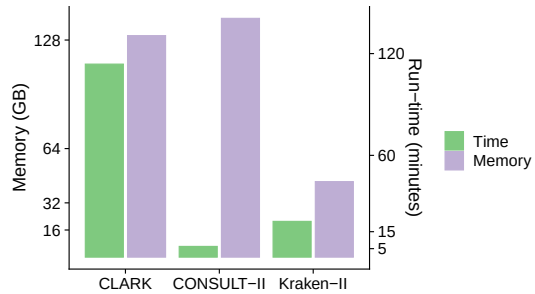

Fig. 5: Run-time and peak memory consumption for classification of 30 microbial queries, each with 66667 150bp short reads. Computation on a machine with 2.2 GHz CPU using 24 threads.

repetitive database loading (per query file), whereas CONSULT-II analyzes all queries in a single run. CONSULT-II has the highest memory footprint (140Gb), followed by CLARK (113Gb). Kraken-II has much better memory efficiency, using only 44Gb.

Discussion

We introduced CONSULT-II for taxonomic identification 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.

References

Ames, S. K. *et al.* (2013). Scalable metagenomic taxonomy classification using a reference genome database. *Bioinformatics (Oxford, England)*, **29**(18), 2253–2260.

Asnicar, F. *et al.* (2020). Precise phylogenetic analysis of microbial isolates and genomes from metagenomes using PhyloPhlAn 3.0. *Nature Communications*, **11**(1), 2500. Publisher: Springer US ISBN: 0348556977.

Berlin, K. *et al.* (2015). Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nature biotechnology*, **33**(6), 623–630.

Blanke, M. and Morgenstern, B. (2020). Phylogenetic placement of short reads without sequence alignment. *bioRxiv*, pages 2020–10. Publisher: Cold Spring Harbor Laboratory.

Brown, D. and Truszkowski, J. (2013). LSHPlace: Fast phylogenetic placement using locality-sensitive hashing. In *Pacific Symposium On Biocomputing*, pages 310–319. ISSN: 2335-6936.

Buhler, J. (2001). Efficient large-scale sequence comparison by locality-sensitive hashing. *Bioinformatics*, **17**(5), 419–428.

Choi, J. *et al.* (2017). Strategies to improve reference databases for soil microbiomes. *The ISME Journal*, **11**(4), 829–834.

Handelsman, J. (2004). Metagenomics: application of genomics to uncultured microorganisms. *Microbiology and molecular biology reviews*, **68**(4), 669–85.

Huang, W. *et al.* (2012). ART: A next-generation sequencing read simulator. *Bioinformatics*, **28**(4), 593–594.

Lau, A.-K. *et al.* (2019). Read-SpaM: assembly-free and alignment-free comparison of bacterial genomes with low sequencing coverage. *BMC Bioinformatics*, **20**(S20), 638.

Liang, Q. *et al.* (2020). DeepMicrobes: taxonomic classification for metagenomics with deep learning. *NAR Genomics and Bioinformatics*, **2**(1).

Liu, B. *et al.* (2011). MetaPhyler: Taxonomic profiling for metagenomic sequences. In *Bioinformatics and Biomedicine (BIBM), 2010 IEEE International Conference on*, pages 95–100. IEEE.

Locey, K. J. and Lennon, J. T. (2016). Scaling laws predict global microbial diversity. *Proceedings of the National Academy of Sciences*, **113**(21), 5970–5975.

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

Luo, Y. *et al.* (2019). Metagenomic binning through low-density hashing. *Bioinformatics*, **35**(2), 219–226.

McDonald, D. *et al.* (2023). Greengenes2 enables a shared data universe for microbiome studies. *Nature Biotechnology*, **in press**(-), –.

McIntyre, A. B. R. *et al.* (2017). Comprehensive benchmarking and ensemble approaches for metagenomic classifiers. *Genome Biology*, **18**(1), 182.

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

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

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

Ondov, B. D. *et al.* (2016). Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biology*, **17**(1), 132.

Ounit, R. *et al.* (2015). CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC Genomics*, **16**(1), 236.

Pachiadaki, M. G. *et al.* (2019). Charting the Complexity of the Marine Microbiome through Single-Cell Genomics. *Cell*, **179**(7), 1623–1635.e11.

Parks, D. H. *et al.* (2020). A complete domain-to-species taxonomy for Bacteria and Archaea. *Nature Biotechnology*, **38**(9), 1079–1086.

Rachtman, E. *et al.* (2020). The impact of contaminants on the accuracy of genome skimming and the effectiveness of exclusion read filters. *Molecular Ecology Resources*, **20**(3), 1755–0998.13135.

Rachtman, E. *et al.* (2021). CONSULT: accurate contamination removal using locality-sensitive hashing. *NAR Genomics and Bioinformatics*, **3**(3), 10.1101/2021.03.18.436035.

Rasheed, Z. *et al.* (2013). 16S rRNA metagenome clustering and diversity estimation using locality sensitive hashing. *BMC Systems Biology*, **7**(Suppl 4), S11.

Şapcı, A. O. B. *et al.* (2023). Consult-ii: Taxonomic identification using locality sensitive hashing. In K. Jahn and T. Vinař, editors, *Comparative Genomics*, pages 196–214, Cham. Springer Nature Switzerland.

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

Segata, N. *et al.* (2012). Metagenomic microbial community profiling using unique clade-specific marker genes. *Nature Methods*, **9**(8), 811–814. ISBN: 1548-7091.

Shah, N. *et al.* (2021). TIPP2: metagenomic taxonomic profiling using phylogenetic markers. *Bioinformatics*, **37**(13), 1839–1845.

Sunagawa, S. *et al.* (2013). Metagenomic species profiling using universal phylogenetic marker genes. *Nature Methods*, **10**(12), 1196–1199.

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

von Meijenfeldt, F. A. B. *et al.* (2019). Robust taxonomic classification of uncharted microbial sequences and bins with CAT and BAT. *Genome Biology*, **20**(1), 217.

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

Wu, S. *et al.* (2020). GMrepo: a database of curated and consistently annotated human gut metagenomes. *Nucleic Acids Research*, **48**(D1), D545–D553.

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

Zhu, Q. *et al.* (2019). Phylogenomics of 10,575 genomes reveals evolutionary proximity between domains Bacteria and Archaea. *Nature Communications*, **10**(1), 5477.

Zhu, Q. *et al.* (2022). Phylogeny-Aware Analysis of Metagenome Community Ecology Based on Matched Reference Genomes while Bypassing Taxonomy. *mSystems*, **7**(2), e0016722.