nature methods

Brief Communication

https://doi.org/10.1038/s41592-024-02273-y

Metabuli: sensitive and specific metagenomic classification via joint analysis of amino acid and DNA

Received: 14 June 2023

Accepted: 11 April 2024

Published online: 20 May 2024

Check for updates

Jaebeom Kim[1] & Martin Steinegger[1,2,3,4]

Metagenomic taxonomic classifiers analyze either DNA or amino acid (AA) sequences. Metabuli (https://metabuli.steineggerlab.com), however, jointly analyzes both DNA and AA to leverage AA conservation for sensitive homology detection and DNA mutations for specific differentiation of closely related taxa. In the Critical Assessment of Metagenome Interpretation 2 plant-associated dataset, Metabuli covered 99% and 98% of classifications of state-of-the-art DNA- and AA-based classifiers, respectively.

Metagenomic studies employ taxonomic classifiers to identify the origin of genetic sequences taken from environments[1]. However, current tools face a dilemma between comparing DNA or six-frame-translated amino acid (AA) sequences. DNA-based classifiers are superior at distinguishing well-studied clades but inferior at detecting understudied clades, and vice versa for AA-based tools[1,2]. Metabuli solved the dilemma by jointly analyzing both sequences to harness the strengths of both types. In benchmarks favoring either DNA- or AA-based analysis, Metabuli consistently demonstrated proficiency, unlike other tools with fluctuating performance due to their inherent dilemmas.

Taxonomic classifiers identify matches between reads and references, through local alignments[3] or, for faster processing, *k*-mer matches of fixed[4,5] or flexible lengths[6,7]. Additionally, approximate mapping is utilized for long-read classification[8]. Despite the variety, current classifiers search the similarity at either the DNA or AA level, facing a dilemma between contrasting capabilities: (1) specificity to resolve between well-studied clades and (2) sensitivity to detect understudied clades using their relatives in the database. DNA-based classifiers exploit DNA mutations for specificity, while AA-based tools capitalize on AA conservation for sensitivity.

Given that metagenomes are mixtures of well- and understudied taxa, the dilemma inevitably restricts full sample characterization. Thus, an ad hoc hybrid approach is frequently employed, wherein reads are first analyzed by a DNA-based classifier, then those unclassified undergo AA-based classification[1,2]. Therefore, we also benchmarked 'Hybrid', an optimal combination of the best-performing DNA- and AA-based classifiers, against Metabuli in each test.

In contrast, Metabuli offers a robust solution by jointly analyzing DNA sequences and their AA translations utilizing metamers (Fig. 1 and Supplementary Fig. 1), novel *k*-mers that jointly encode DNA and AA information, requiring only two-thirds the storage compared to separate encodings. During database creation, Metabuli employs Prodigal[9] to predict reference genome's open reading frames (ORFs) and extends them to include intergenic regions, which is overlooked when relying solely on coding sequences[1]. Since metamers are codon based, storing them requires 1/3 of the genome length, and further compression is achieved by storing only distinct metamers within a species. Despite storing both DNA and AA, Metabuli's database was only about 1.5 times larger than Kraken2's probabilistic database (Extended Data Table 1).

During classification, Metabuli compares query metamers against the reference to find exact AA matches for sensitivity, using their DNA information for specificity in three ways: (1) removing too distant matches, (2) defining overlaps when assembling continuous matches per species, and (3) scoring the assemblies (Supplementary Figs. 2 and 3). Each read is assigned to the highest-scoring species, or to the lowest common ancestor (LCA) of equally highest-scoring species. In this process, Metabuli-P (precision mode) uses score thresholds to

[1]Interdisciplinary Program in Bioinformatics, Seoul National University, Seoul, Republic of Korea. [2]School of Biological Sciences, Seoul National University, Seoul, Republic of Korea. [3]Institute of Molecular Biology and Genetics, Seoul National University, Seoul, Republic of Korea. [4]Artificial Intelligence Institute, Seoul National University, Seoul, Republic of Korea. e-mail: martin.steinegger@snu.ac.kr

https://doi.org/10.1038/s41592-024-02273-y

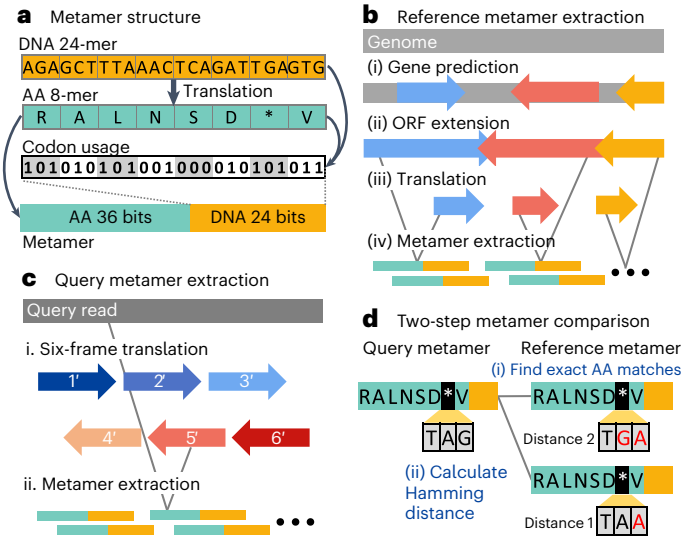


Fig. 1 | Metabuli's workflow. a, The 8-mer translation is encoded using 36 bits, and its codon usage is recorded to store DNA information. Since an AA is coded by maximally six codons, three bits per AA are sufficient to indicate each case. **b**,**c**, Reference (**b**) and query (**c**) metamer extraction process. **d**, Exact AA matches are searched, for which DNA Hamming distances are calculated.

reduce false positive (FP) and overconfident classifications (Methods and Supplementary Figs. 4–7). Metabuli classified 15 million Illumina reads in 10 min and 875,545 PacBio HiFi long reads in 20 min, compared with Kraken2's 24 and 76 s (Extended Data Table 1). Notably, Metabuli operates within user-specified random-access memory (RAM) limits to search any database fitting in storage, even completing these tasks on a 8 GiB RAM notebook.

In this Brief Communication, we conducted three sets of synthetic benchmarks based on Genome Taxonomy Database (GTDB)[10], using short and long reads simulated across four sequencing platforms (Extended Data Fig. 1). The first and second sets, subspecies inclusion and subspecies exclusion tests, evaluated subspecies- and species-level performance by including either the queried subspecies or their siblings in the database, respectively (Fig. 2a). On both sets, DNA-based tools and Metabuli outperformed AA-based classifiers, especially on Illumina reads with a recall 0.2–0.3 higher, underscoring their superior ability to make specific classifications. Hybrid's performance was determined by its DNA-based tool, which left little room for AA-based improvement.

However, in the third set (species exclusion tests) where queried species were excluded from the database, DNA-based classifiers were surpassed by AA-based ones in detecting genus-level homology (Fig. 2b), having a recall 0.2 lower on Illumina reads. Hybrid also performed worse than its AA-based tool as its DNA-based tool misclassified reads that the AA-based tool could have accurately classified. In contrast, Metabuli secured top-level F1 scores, standing out as the only method consistently competent across all test sets. In addition, Metabuli-P achieved the highest or second-highest precision, maintaining similar F1 scores. Moreover, despite using AA information, Metabuli was robust to frame-shifting errors in non-HiFi long reads, unlike AA-based Kaiju and MMseqs2, by incorporating matches from multiple frames[11].

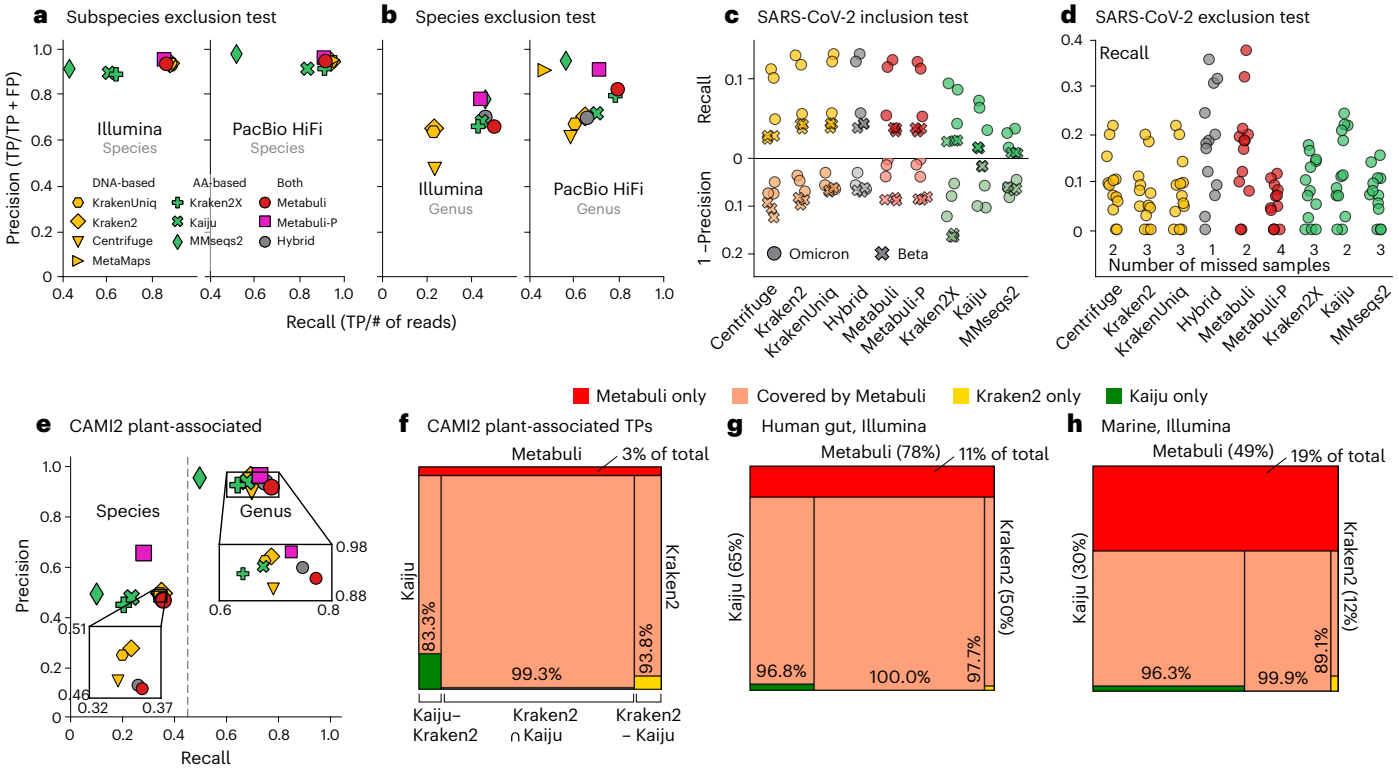


Fig. 2 | Benchmark results. Hybrid = (x, y): best-performing DNA-based x and AA-based y. **a**, Synthetic subspecies exclusion test. Siblings, not queried subspecies, were in the database. Hybrid = (KrakenUniq, Kraken2X). **b**, Synthetic species exclusion test. Siblings, not queried species, were in the database. Illumina Hybrid = (Kraken2, MMseqs2), PacBio HiFi Hybrid = (Kraken2, Kraken2X). **c**, SARS-CoV-2 inclusion test. Analysis of RNA-seq samples from patients with COVID-19. Each circle indicates measured performance on a sample. The reference included five SARS-CoV-2 variants. Patients with Omicron or Beta variant were queried. Classifications to correct (incorrect) variants counted as true (false) positives. Hybrid = (KrakenUniq, Kraken2X). **d**, SARS-CoV-2 exclusion test. Analysis of RNA-seq samples from patients with COVID-19. Each circle indicates measured performance on a sample. SARS-CoV-2 was excluded from reference. Hybrid = (Centrifuge, Kaiju). **e**,**f**, CAMI2 plant-associated reads. GTDB genomes and the CAMI2-provided taxonomy were used for database creation (**e**); Venn diagram of genus-level TPs (**f**). **g**,**h**, Real metagenomes from human gut (**g**) and marine environment (**h**). Venn diagram of classified reads. Classified proportion of each tool in parentheses. In **f**–**h**, the area is proportional to the number of reads.

We also performed two primary pathogen detection benchmarks: strain identification and emerging pathogen discovery (Fig. 2c,d). We constructed databases that either included or excluded severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and its variants for these respective tests. In the inclusion test, DNA-based classifiers and Metabuli detected more variant-specific reads from RNA sequencing (RNA-seq) data from patients with coronavirus disease 2019 (COVID-19) than AA-based tools, with KrakenUniq determining Hybrid's performance. In the exclusion test, the LCA of SARS-CoV-1 and SARS-CoV-2, identified as SARS-related coronavirus species, was counted as true positive (TP) taxon. Although AA-based tools classified more reads to the genus of the culprit lineage, the number of their classifications to the TP species was similar to that of DNA-based tools. Hybrid surpassed its DNA-based tool, with its AA-based tool capturing signals from the initially unclassified reads. Remarkably, Metabuli matched Hybrid's performance, effectively bridging the gap between DNA- and AA-based classifications.

To reflect real-world scenarios, we utilized datasets from Critical Assessment of Metagenome Interpretation 2 (CAMI2 (ref. 12)), which contain organisms with varying query-to-database distances (Extended Data Fig. 2). DNA-based classifiers outperformed AA-based ones in the strain-madness dataset, but their performance became gradually similar in the marine and plant-associated datasets as query-to-database distances grew. For the plant-associated, Metabuli and Metabuli-P achieved the highest recall and precision, respectively (Fig. 2e). When Metabuli's genus-level TP set were compared with those of Kaiju and Kraken2, the top-performing AA-based and DNA-based tools, respectively, Metabuli covered 99.3% of Kaiju ∩ Kraken2, 83.3% of Kaiju − Kraken2 and 93.8% of Kraken2 − Kaiju, additionally classifying a unique 3% of the total reads (Fig. 2f).

Next, we conducted similar analysis using real metagenomes. Since real reads lack ground-truth labels, we compared the proportion of reads classified by each tool. For the human gut short-read sample, Kaiju and Kraken2 classified 65% and 50% of the reads, respectively, but this decreased to 30% and 12% as query-to-database distance increased in the marine data (Fig. 2g,h). This decline with marine data was also observed in Oxford Nanopore Technologies (ONT) data but not in PacBio HiFi data (Extended Data Fig. 3). Notably, for human gut and marine short-read data, Metabuli covered >96% of Kaiju − Kraken2, 89–98% of Kraken2 − Kaiju and >99% of Kaiju ∩ Kraken2, uniquely classifying an additional 11% and 19% of the respective samples.

Metabuli still faces some challenges. Like other methods, Metabuli's subspecies-level resolution decreases as more query-related subspecies are present in the database (Extended Data Fig. 4)[13]. Moreover, compared with long-read specialized MetaMaps, Metabuli had lower subspecies-level resolution for long reads (Extended Data Fig. 1b–d). Finally, Metabuli-P's species-level threshold wasn't suitable for viruses with considerable diversity within a species (Fig. 2d).

Despite these limitations, only Metabuli showed robust performance across all benchmarks. DNA-based classifiers excelled at distinguishing between closely related known organisms but struggled in detecting novel species, whereas AA-based methods showed the opposite property. Hybrid approaches generally showed improvements but were undermined when the initially applied tool made many false classifications. Importantly, the optimal hybrid combination, as examined here, requires knowledge of the best-performing tools, which is impossible with real data lacking true labels, rendering it less practical.

In summary, Metabuli is the most suitable for metagenomic classification, where samples contain a mixture of well- and understudied species. By integrating DNA- and AA-based analysis, Metabuli effectively identified both, highlighting its potential to transform clinical diagnostics and ecological studies.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41592-024-02273-y.

References

1. Simon, H. Y., Siddle, K. J., Park, D. J. & Sabeti, P. C. Benchmarking metagenomics tools for taxonomic classification. *Cell* **178**, 779–794 (2019).
2. Nooij, S., Schmitz, D., Vennema, H., Kroneman, A. & Koopmans, M. P. Overview of virus metagenomic classification methods and their biological applications. *Front. Microbiol.* **9**, 749 (2018).
3. Mirdita, M., Steinegger, M., Breitwieser, F., Söding, J. & Levy Karin, E. Fast and sensitive taxonomic assignment to metagenomic contigs. *Bioinformatics* **37**, 3029–3031 (2021).
4. Wood, D. E., Lu, J. & Langmead, B. Improved metagenomic analysis with Kraken 2. *Genome Biol.* **20**, 1–13 (2019).
5. Breitwieser, F. P., Baker, D. N. & Salzberg, S. L. Krakenuniq: confident and fast metagenomics classification using unique *k*-mer counts. *Genome Biol.* **19**, 198 (2018).
6. Menzel, P., Ng, K. L. & Krogh, A. Fast and sensitive taxonomic classification for metagenomics with Kaiju. *Nat. Commun.* **7**, 11257 (2016).
7. Kim, D., Song, L., Breitwieser, F. P. & Salzberg, S. L. Centrifuge: rapid and sensitive classification of metagenomic sequences. *Genome Res.* **26**, 1721–1729 (2016).
8. Dilthey, A. T., Jain, C., Koren, S. & Phillippy, A. M. Strain-level metagenomic assignment and compositional estimation for long reads with metamaps. *Nat. Commun.* **10**, 3066 (2019).
9. Hyatt, D. et al. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinform.* **11**, 119 (2010).
10. Parks, D. H. et al. GTDB: an ongoing census of bacterial and archaeal diversity through a phylogenetically consistent, rank normalized and complete genome-based taxonomy. *Nucleic Acids Res.* **50**, D785–D794 (2022).
11. Watson, M. & Warr, A. Errors in long-read assemblies can critically affect protein prediction. *Nat. Biotechnol.* **37**, 124–126 (2019).
12. Meyer, F. et al. Critical assessment of metagenome interpretation: the second round of challenges. *Nat. Methods* **19**, 429–440 (2022).
13. Nasko, D. J., Koren, S., Phillippy, A. M. & Treangen, T. J. Refseq database growth influences the accuracy of *k*-mer-based lowest common ancestor species identification. *Genome Biol.* **19**, 1–10 (2018).

Methods

Benchmarked software

We included four DNA-based classifiers in our benchmarks: KrakenUniq (v0.7.3)[5], Kraken2 (v2.1.2)[4], Centrifuge (v1.0.4)[7] and MetaMaps (v0.1)[8]. KrakenUniq is an improved version of a *k*-mer-based classifier, Kraken, with the functionality of considering the coverage of unique *k*-mers of each species to reduce FP classification. Kraken2 is a faster and lighter version of Kraken, introducing a probabilistic database and the minimizer concept. For Kraken2, --minimum-hit-groups was set as 3 following a recommended usage[14]. Centrifuge, another widely used tool, also searches for *k*-mer matches, but its usage of FM-index allows it to find k-mer matches of flexible lengths. We ran Centrifuge with the '-k 1' option to report at most one classification per read. MetaMaps is an approximate mapping-based long read classifier. Its results for each read include mapping qualities for each of the reference sequences to which the read was mapped. Then, each read is classified to the taxon with the highest mapping quality. Classification results are reported after refinement through an expectation–maximization (EM) process. In Fig. 2a,b and Extended Data Fig. 1, MetaMaps_EM represents the performance measured on the EM-processed results. To discern the performance improvement attributable to the EM step, we developed the mapping2taxon module in Metabuli. This module takes the mapping result of MetaMaps and classifies each read to the taxon with the highest mapping quality. In case several taxa have equal mapping qualities, the read is classified to the LCA of these taxa. Its performance is denoted as MetaMaps.

In the case of AA-based classifiers, we benchmarked Kaiju (v1.9)[6], Kraken2X (v2.1.2) and MMseqs2 Taxonomy (v13.45111)[3]. Kaiju, a widely used AA-based classifier, is also based on FM-index like Centrifuge but searches for AA sequence matches. Kraken2X, the translated search mode of Kraken2, was included in our tests because comparing it with Kraken2 allows focusing on the effect of the utilized sequence type. MMseqs2 Taxonomy classifies reads by calculating local alignments to references and was included in the benchmarks as an AA-based classifier designed for both short and long reads. For short reads, we applied '--orf-filter 0' as recommended in MMseqs2's manual.

We included an approach, denoted as 'Hybrid' in the benchmarks by first running the best-performing DNA-based classifier and then applying the best-performing AA-based tool to reads unclassified by the initial DNA-based tool[2]. To solely assess effect of using both DNA- and AA-based tools, we excluded MetaMaps_EM to avoid measuring the effect of post-processing refinement. Additionally, we tested the less commonly used reverse order, applying the AA-based one first, in synthetic benchmarks (Supplementary Tables 1 and 3) and pathogen detection tests (Supplementary Tables 5 and 6). However, this approach failed to show improvement over its AA-based tool alone or performed worse than the Hybrid configuration.

Simulated read generation

To simulate paired-end short reads used in synthetic benchmarks, we used the mason_simulator module of Mason2 (v2.0.9)[15]. The reads were 150 nt in length and included simulated errors at rates of 0.11% for mismatches, 0.005% for insertions and 0.005% for deletions. These error rates were based on the performance of the NovaSeq 6000 sequencer. As with Mason2's default settings, the mismatch probability at the beginning and end of the reads was set to 0.5% and 0.22%, respectively. When provided to MMseqs2 Taxonomy, simulated reads were concatenated as it does not support paired-end reads. We inserted 'NN' between each paired reads to break MMseqs2's *k*-mer extraction at the border. In the case of long reads, we used PBSIM3 (v3.0.0)[16] to simulate reads of Oxford Nanopore Technologies with options '--strategy wgs --method errhmm --errhmm ERRHMM-ONT.model --depth 3', and for PacBio Sequel, '--strategy wgs --method errhmm --errhmm ERRHMM-SEQUEL.model --depth 3'. We simulated PacBio HiFi (Sequel-CCS) reads following the manual of PBSIM3. First we used PBSIM3 to simulate multi-pass Sequel reads with options '--strategy wgs --method errhmm --errhmm ERRHMM-ONT.model --depth 1 --pass-num 10' and then piped the results to PacBio's CCS-read generation tool.

GTDB genomes

The GTDB was used for several benchmarks as well as for the calibration of Metabuli-P as it provides phylogenetically consistent taxonomy based on genomic distance measures. For these, we started with a subset of GTDB R202 consisting of 258,406 genomes from 47,894 species clusters. We used the GTDB_metadata_filter.R module in the pipeline Struo v0.1.7 (ref. 17) to obtain a list of 22,973 genomes that were assembled at the level of complete genome or chromosome, had CheckM completeness >90 and had CheckM contamination <5. The filtered genomes were downloaded using Struo's genome_download.R module, and 22,819 successfully downloaded genomes of 6,186 species were used. NCBI-style taxonomy dump files for the GTDB were generated by gtdb_to_taxdump[18] module. The proteome corresponding to each genome was computed by Prodigal (v2.6) with default settings.

Metabuli: database creation

Metabuli builds a reference database of computed metamers from nucleotide sequences following the procedure below (Supplementary Fig. 1a–e).

ORF prediction and extension. Metabuli utilizes Prodigal for ORF prediction in reference sequences. To enhance the prediction process' efficiency, we implemented three optimizations. (1) Metabuli stores reference sequences in separate FASTA files for each species, then it trains Prodigal once for each species using the longest sequence in each FASTA file before predicting genes. This approach reduces training time, considering the presence of multiple assemblies for a single species. (2) We narrowed down the calculation range of Prodigal's dynamic programming during both training and prediction steps. It may cause Prodigal to miss very long genes. However, when tested on the *Escherichia coli* genome, it produced the same results as the original Prodigal, while reducing runtime by half. (3) We parallelized the training and prediction processes by distributing jobs for each species across multiple threads. After gene prediction, genes that are fully nested in longer ones are removed. The ORFs of the remaining genes are extended to cover all intergenic regions while maintaining the predicted reading frame.

Reference metamer calculation and compression. Metabuli computes reference metamers from the extended ORFs and their translations. First, it translates the extended ORFs based on the frame predicted by Prodigal. In this way, the entire reference sequence undergoes translation once, using different frames in different regions. Next, Metabuli calculates metamers (Supplementary Fig. 1, Metamer) using DNA information from the original DNA sequence and AA information from the extended ORF translations. Metabuli utilizes TANTAN[19] to mask low-complexity regions at the DNA level with repeat probability >0.9 and excludes the masked regions from metamer calculation. All computed metamers are sorted numerically and then by their associated species ID. Because metamers encode AAs in the high-order bits, metamers encoding the same AA sequence are placed consecutively after sorting, and within them, they are grouped by codon usage, followed by their associated species ID. Then, redundant metamers from the same species are removed, retaining only one of them (Supplementary Fig. 1c). The reduced metamer list is then further compressed as follows (Supplementary Fig. 1d). The full numerical value of the first metamer is stored. For all other metamers on the list, only the increment value from the previous metamer is stored. The 64-bit encoding of the first metamer and the increments are then scanned as four slices of 15 bits each (the last 4 bits are unused). The slice of the lowest-order bits and any slice where some of the bits are turned on are copied and

stored in 16 bits with one extra bit for an end flag. The end flag indicates whether the copied slice was the last one to be saved from a specific 64-bit value (where 1 = the copied slice is the last one). The optimal case is when only one slice is stored per metamer, yielding a compression ratio of 4. The more reference metamers there are, the smaller the increments between consecutive ones tend to be, so the compression rate becomes closer to 4. For example, when Metabuli was used to create a database from genomes of NCBI RefSeq release 217 (~1.1 TB), the compression rate was about 3. Throughout this procedure, the reference sequence ID associated with each metamer is stored alongside it as well as information concerning metamer redundancy.

Genetic code. Metabuli uses the standard genetic code when calculating metamers from query reads because their taxonomic origin is generally unknown in metagenomic samples. To best match query metamers, Metabuli also uses the standard genetic code while extracting reference metamers from reference genomes. By adhering to a single genetic code, we could make the codon encoding of the metamer compact and use it to index a Hamming distance lookup table ('Calculating Hamming distance' section and Supplementary Fig. 2).

For ORF prediction, Metabuli utilizes Prodigal to process all reference sequences using the genetic code 11, which is Prodigal's default setting and is the extension of the standard code with an additional accounting for variations in start codons of bacteria, archaea and prokarote viruses. During development, we performed subspecies and species exclusion tests to compare Metabuli's performance on the genera *Escherichia* and *Mycoplasma*, whose native genetic codes are 11 and 4, respectively. Metabuli correctly classified over 99.8% of reads for both genera in the subspecies exclusion test. In the species exclusion test, 74–88% and 75–83% of reads were accurately classified for each genus, respectively. This trivial difference in Metabuli's performance supported using Prodigal's default setting for all species.

Metabuli: database decompression and usage

The values of the first metamer and the increments can be computed back from the stored compression by concatenating corresponding slices in a 64-bit data type. From the second metamer, their values are sequentially calculated by summing up each increment.

Metabuli: classification

Metamer match search. Query metamers from reads are sorted and compared with the reference metamer list to find matches (Supplementary Fig. 1f,g). Because both query and reference metamers are sorted, a single iteration through the lists is enough to find all matches.

Calculating Hamming distance. After a query metamer is matched with reference metamers that are identical to it on the AA level, the DNA Hamming distance of each match is calculated using a Hamming distance lookup table (Supplementary Fig. 2). In this table, the 3-bit representations of any pair of synonymous codons are used as indices to retrieve their distance. For each query metamer, the minimum Hamming distance (minH) is searched to filter out matches with Hamming distance >MIN(7, 2 × minH). We allow matches to have two times the minimum distance because matches from the true taxon do not always have the smallest distance. Matches with Hamming distance >7 are filtered out to guarantee matches have DNA identity >70%.

Computing sequence similarity and assigning taxonomy. The matches that passed the Hamming distance criteria were grouped by species and query translation frame and sorted by their coordinates on the read. For each species, overlap graphs of matches are constructed (Supplementary Fig. 3) for each query translation frame. Each match is a node, and edges connect consecutive matches. Two matches are considered consecutive when (1) their query metamers are extracted from positions that differ by three nucleotides in the same reading frame and (2) their DNA sequences within the overlapping region are identical.

For each starting node (match) of consecutive matches, a path that traverses from the start to a certain terminal node with the highest score is searched in a depth-first way (Supplementary Fig. 3). The score of each path is calculated as follows. An AA match results in 3, 1.5, 1 or 0.5 points, respectively, when the Hamming distance is 0, 1, 2 or 3. The points of eight AA matches of a starting node are summed up and passed to its next matches. From the next matches, the score of each added AA match is summed up cumulatively to calculate the score of paths to terminal matches.

After the best paths for each starting match are selected, they are combined in a greedy way to cover the query read. A path with the highest score is included first, and nonoverlapping next-best matches are included until no path can be added. Exceptionally, overlaps smaller than 24 nt are allowed because such cases can happen when the matched reference metamers are extracted from a border of different ORFs. The total score of the combined paths becomes a score of the species.

Metabuli assigns the read to the species of the highest score, and the score is divided by the query length to get the sequence identity score of this classification. Within the selected species, lower-taxon specific metamer matches are used for lower-level classifications.

Metabuli: Metabuli-P

Notably, as with other short *k*-mer-based classifiers, relying on few matches can often lead to FP or overconfident classifications. FP classification occurs mainly when the matched region is short. The similarity between a pair of sequences is expected to be higher if the pair belongs to the same lower taxonomic rank (rather than a higher rank). Overconfidence occurs when a read is classified at lower ranks like species or subspecies with not enough sequence similarity. To address this, Metabuli's precision mode (Metabuli-P) uses two sequence similarity thresholds to avoid false and overconfident classifications. These thresholds were set on the basis of similarity score distributions within prokaryotic and viral genera and species (Supplementary Figs. 4–7).

Distribution of sequence similarity scores. We investigated the distribution of sequence similarities underlying TP and FP classifications using prokaryotes and viruses. Prokaryotic and viral species were identified on the basis of two criteria: (1) there was at least one other species belonging to the same genus in the database, and (2) the database contained genomes of at least two of their subspecies. For prokaryotes, we could find 435 species, from the 22,819 GTDB genomes, that met the two criteria. We then designed two settings: subspecies exclusion and species exclusion. In the subspecies exclusion, for each of the 435 species, one subspecies was included in the reference database while one of its sibling subspecies was excluded from it and used to simulate query reads. In the species exclusion, the same database was used, and for each of the 435 species a random sibling species from the same genus was used to generate query reads. In both settings, 45,000 paired-end short reads for each query genome were simulated using Mason2. For long reads, 3× depth of ONT and PacBio Sequel II reads and 1× of PacBio HiFi reads were simulated using PBSIM3.

In the case of viruses, we used NCBI taxonomy and Viral RefSeq. We could not find enough viral species fulfilling both criteria. Therefore, for the subspecies exclusion setting, we applied the second criterion to find 211 species with at least two subspecies. In the case of the species exclusion setting, the first criterion was applied to find 889 genera that have at least two species. In both settings, 10,000 paired-end reads were simulated from each query genome. A 3× depth of ONT and PacBio Sequel II reads and 1× of PacBio HiFi reads were simulated using PBSIM3. Then, we used Metabuli to classify query reads in the various test settings and examined the sequence similarity scores underlying the TP or FP classifications.

Determining thresholds. Examination of the sequence similarity distributions of Illumina short reads revealed that FP's relative frequency peaks under sequence similarity of 0.1 (Supplementary Fig. 4). Furthermore, the vast majority (88.6–99.6%) of all TPs are associated with a sequence similarity score greater than 0.15 (Supplementary Fig. 4a–d), while many FPs (27.0.3–50.4%) are associated with a lower score (Supplementary Fig. 4e–h). Therefore, Metabuli-P for short reads is set to leave a query as unclassified if its best genus-level similarity score is lower than 0.15. In the subspecies exclusion settings, 96.7% (prokaryote) and 81.6% (virus) of the TPs are associated with a similarity score greater than 0.5 while only 15.4% (prokaryote) and 59.1% (virus) of the FPs scored as high. Thus, Metabuli-P is set to classify a read at the species level or a lower rank only if it has a similarity score of >0.5 to at least one species. We performed the same examination for simulated long reads. For PacBio HiFi long reads, the minimum scores for classification and species-level classification were set as 0.07 and 0.3, respectively (Supplementary Fig. 5). In the case of ONT and PacBio Sequel II (Supplementary Figs. 6 and 7), the minimum scores for classification were set to 0.008 and 0.005, respectively, but a proper threshold for species-level classification was not identified.

Synthetic benchmarks

We conducted three sets of synthetic benchmarks: subspecies inclusion, subspecies exclusion and species exclusion test. Each set had different query/reference sequences and measured performance at different ranks according to their taxonomic relationships, with the LCA of each query and its closest sibling in the reference regarded as the true label of the query. When measuring at a certain rank, unclassified reads as well as reads classified at higher ranks were considered as false negatives to penalize less informative classifications. Classifications to clades containing the true label were counted as TPs, and to others as FPs.

Subspecies inclusion test. We examined the 22,819 complete genome or chromosome level assemblies in the GTDB by their species and identified 1,626 species that had at least two subspecies with a genome in the database (26% of all species in the GTDB). Of these, 435 species were used for the score threshold setting of Metabuli-P (Supplementary Fig. 4). The remaining 1,191 species (19% of all species in the filtered and downloaded GTDB genomes) contributed 2 subspecies each, from which 6,150 paired-end reads were simulated with Mason2 (~15M reads in total). Each of these genomes was also used to simulate 3× depth ONT and PacBio Sequel II reads and 1× PacBio HiFi long reads using PBSIM3. Performance metrics were measured at subspecies, species, genus and family ranks. The genomes used for database creation and read simulation are listed in Supplementary Table 4.

Subspecies exclusion test. One subspecies from each of the 1,191 species used in the subspecies inclusion test, was randomly excluded from the reference database. These 1,191 excluded subspecies were then utilized to simulate query reads. We employed Mason2 to simulate short (150 nt) reads, producing 12,300 paired-end reads per excluded subspecies, resulting in a total of approximately 15 million reads. Additionally, long 3× depth PacBio Sequel II, 3× depth ONT and 1× depth PacBio HiFi reads were simulated using PBSIM3. Evaluations were conducted at the species and three higher taxonomic ranks. The assemblies for database creation and read simulation are listed in Supplementary Table 4.

Species exclusion test. The 22,819 GTDB genomes were examined by their genera. We identified 802 genera which had at least two species with a genome in the database. Of these, 435 were used for the score threshold setting of Metabuli-P (Supplementary Fig. 4). The remaining 367 genera were used for the exclusion test. In this setting, ~50,000 reads were simulated from each species (~20M reads in total) using Mason2. PBSIM3 was used to simulate ONT, PacBio Sequel II and PacBio HiFi reads of the species. Performance was measured at genus and three higher taxonomic ranks. In Supplementary Table 4, the assembly accessions of the genomes for database creation and read simulation are provided.

Pathogen detection tests

SARS-CoV-2 inclusion test. Reference databases were built using viral genomes or proteomes of NCBI RefSeq (release 212), T2T-CHM13v2.0 and five SARS-CoV-2 variants: Alpha (OK244698.1), Beta (OK238749.1), Delta (OM793985.1), Gamma (OL565980.1) and Omicron (ON235252.1). We manually included these variants as children of SARS-CoV-2 in the NCBI taxonomy database. Two sets of RNA-seq data from patients with COVID-19 were used as query reads. One set was prepared from three patients infected by the Beta variant[20], and the other from three patients infected by the Omicron variant[21]. The accessions for public data used as query and the raw data of the performance measure of each tool are available in Supplementary Table 5.

SARS-CoV-2 exclusion test. The database for each tool was constructed using the taxonomy of NCBI and the viral genomes or proteomes of RefSeq (release 212) and T2T-CHM13v2.0, excluding all SARS-CoV-2 sequences. Due to this exclusion, SARS-CoV-1 is the closest relative in the reference database to any variant of SARS-CoV-2. RNA-seq data from 13 patients with SARS-CoV-2 and five controls prepared in a host-response study were used as query reads[22]. Classifications as SARS-related coronavirus are counted as TPs for patient samples and FPs for control samples, respectively. For controls, all classifiers did not make any FPs (Supplementary Table 6). The estimated number of SARS-CoV-2 reads in each sample was calculated by multiplying the total number of RNA-seq reads by the reads per million (RPM) of reads aligned to the SARS-CoV-2 genome. The RPM values were taken from the original study. The accessions of the queried public data and raw measures of performance of each tool are available in Supplementary Table 6.

CAMI2 benchmarks

We used paired-end reads of strain madness, marine and plant-associated datasets and taxonomy provided in CAMI2 (ref. 12). In the case of CAMI2-provided reference databases for DNA- and AA-based tools (nt and nr), where there is no one-to-one relationship between their entries, it is possible that some taxa may be under- or overrepresented compared with each other. This discrepancy can lead to a potentially unfair comparison between the two groups of classifiers. To replace the CAMI2-provided databases, we used the reference genomes and proteomes used in the prokaryote subspecies inclusion test. These references and a CAMI-provided mapping from accessions to taxonomic IDs were utilized for database creation. Metabuli, Centrifuge and KrakenUniq used 7,318 genomes, which together with two additional genomes were used for Kraken2, Kraken2X, Kaiju and MMseqs2. CAMI2 provides 10, 21 and 100 query samples for the marine, plant-associated and strain-madness benchmarks, respectively. To reduce the runtime of the benchmarks, we took all, every second and every tenth query samples, respectively. We also used the CAMI2-provided ground truth labels for each read. When measuring performance at a certain rank, we disregarded classifications for reads whose ground truth taxon ranked higher than the measurement rank.

Benchmarks with real metagenomes

We compared Metabuli with the best-performing DNA- and AA-based tools for real metagenomic short and long reads. True labels of the reads are not available, so we focused solely on counting how many reads could be classified by each tool. Since this setting is favorable to tools with high sensitivity, we selected the best-performing DNA-based tool and AA-based tool based on their recalls in the synthetic prokaryote

benchmarks: Kraken2 and Kaiju for short reads and Kraken2 and Kraken2X for long reads. There databases were built using GTDB genomes and a human genome (T2T-CHM13v2.0) with GTDB taxonomy edited to include human taxon. We challenged the classifiers on two distinct metagenomes: one obtained from a well-studied environment, specifically human gut samples (SRR24315757 for Illumina reads, SRR15489017 for PacBio HiFi reads[23] and SRR17913199 for ONT reads[24]), and the other from a less-studied environment, a marine sample (SRR23604821 for Illumina reads[25], ERR4920902 for PacBio HiFi reads[26] and SRR24091366 for ONT reads[27]).

Resource measurement

Maximum RAM usage (maximum resident set size) and elapsed time of each tool were measured with the GNU 'time -v' command. The average performance over five repeated measurements is reported in Extended Data Table 1.

Computing resource

For the resource measurement, we used a server and a MacBook Air. The server was equipped with a 64-core AMD EPYC 7742 CPU and 1 TB of RAM, and the MacBook Air (2020) had 8 GB RAM and an Apple M1 chip (8-core CPU).

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The raw data for Fig. 2 and Extended Data Figs. 1–4 are provided as Source data. Performance measures at different ranks of benchmarks of Fig. 2a,b and Extended Data Fig. 1 are available in Supplementary Tables 1–3. The assemblies used for read simulation and database creation in synthetic benchmarks are listed in Supplementary Table 4, and the simulated reads are available via Zenodo at https://doi.org/10.5281/zenodo.10250585 (ref. 28). More detailed results and utilized accessions of Fig. 2c,d are provided in Supplementary Tables 5 and 6. The databases used in Fig. 2c,d were built using viral genomes (release 212) and a human genome (GCF_009914755.1) downloaded from NCBI RefSeq, and accessions of genomes of analyzed SARS-CoV-2 variants were denoted in 'Pathogen detection tests' section in Methods. Performance measures at different ranks of Fig. 2e and Extended Data Fig. 2 are provided in Supplementary Tables 7–9. Precision and recall of Extended Data Fig. 4 are available in Supplementary Table 10. The accessions of real data analyzed in Fig. 2g,h and Extended Data Fig. 3 are denoted in 'Benchmarks with real metagenomes' section in Methods. CAMI2-provided datasets and taxonomy used in Fig. 2e,f and Extended Data Fig. 2 can be downloaded from https://data.cami-challenge.org/participate. Source data are provided with this paper.

Code availability

Metabuli is GPLv3-licensed free open-source software. The source code and ready-to-use binaries, as well as precomputed databases (Supplementary Table 11), can be downloaded at metabuli.steinegerlab.com. The scripts used for benchmarks and plots are available at https://github.com/jaebeom-kim/metabuli-analysis and https://github.com/jaebeom-kim/metabuli-plots.

References

14. Lu, J. et al. Metagenome analysis using the Kraken software suite. *Nat. Protoc.* **17**, 2815–2839 (2022).
15. Holtgrewe, M. *Mason - A Read Simulator for Second Generation Sequencing Data*. Technical Report (FU Berlin, 2010).
16. Ono, Y., Hamada, M. & Asai, K. PBSIM3: a simulator for all types of PacBio and ONT long reads. *NAR Genom. Bioinform.* **4**, lqac092 (2022).
17. de la Cuesta-Zuluaga, J., Ley, R. E. & Youngblut, N. D. Struo: a pipeline for building custom databases for common metagenome profilers. *Bioinformatics* **36**, 2314–2315 (2020).
18. Youngblut, N. & Shen, W. nick-youngblut/gtdb_to_taxdump: Zenodo release. *Zenodo* https://doi.org/10.5281/zenodo.3696964 (2020).
19. Frith, M. C. A new repeat-masking method enables specific detection of homologous sequences. *Nucleic Acids Res.* **39**, e23 (2011).
20. Rahaman, M. M. et al. Genomic characterization of the dominating Beta, V2 variant carrying vaccinated (Oxford-AstraZeneca) and nonvaccinated COVID-19 patient samples in Bangladesh: a metagenomics and whole-genome approach. *J. Med. Virol.* **94**, 1670–1688 (2022).
21. Lentini, A., Pereira, A., Winqvist, O. & Reinius, B. Monitoring of the SARS-CoV-2 Omicron BA.1/BA.2 lineage transition in the Swedish population reveals increased viral RNA levels in BA.2 cases. *Med* **3**, 636–643 (2022).
22. Desai, N. et al. Temporal and spatial heterogeneity of host response to SARS-CoV-2 pulmonary infection. *Nat. Commun.* **11**, 6319 (2020).
23. Gehrig, J. L. et al. Finding the right fit: evaluation of short-read and long-read sequencing approaches to maximize the utility of clinical microbiome data. *Microb. Genom.* **8**, 000794 (2022).
24. Liu, L., Yang, Y., Deng, Y. & Zhang, T. Nanopore long-read-only metagenomics enables complete and high-quality genome reconstruction from mock and complex metagenomes. *Microbiome* **10**, 209 (2022).
25. Barnes, S. J. et al. Metagenome-assembled genomes from photo-oxidized and nonoxidized oil-degrading marine microcosms. *Microbiol. Resour. Announc.* **12,** 6 (2023).
26. Priest, T., Orellana, L. H., Huettel, B., Fuchs, B. M. & Amann, R. Microbial metagenome-assembled genomes of the Fram Strait from short and long read sequencing platforms. *PeerJ* **9**, e11721 (2021).
27. Huang, R. et al. Long-read metagenomics of marine microbes reveals diversely expressed secondary metabolites. *Microbiol. Spectr.* **11**, e0150123 (2023).
28. Kim, J. Simulated query reads used for benchmarks in Metabuli publication. *Zenodo* https://doi.org/10.5281/zenodo.10250585 (2023).

Acknowledgements

The authors thank E. Levy Karin for the valuable scientific feedback and the careful review and revision of the paper; J. Söding for the discussions on metamer encoding; M. Mirdita for the usability improvements of the software; H. Kim for the improvement of figures; S. Jaenicke for the voluntary examination of the software; and M. Kim for the feedback on the paper. M.S. acknowledges support by the National Research Foundation of Korea grants (2020M3-A9G7-103933, 2021-R1C1-C102065 and 2021-M3A9-I4021220), the Samsung DS research fund, and the Creative-Pioneering Researchers Program and AI-Bio Research Grant through Seoul National University.

Author contributions

J.K. and M.S. designed the research, developed the software, performed analysis and wrote the paper.

Competing interests

The authors declare no competing interests.

Additional information

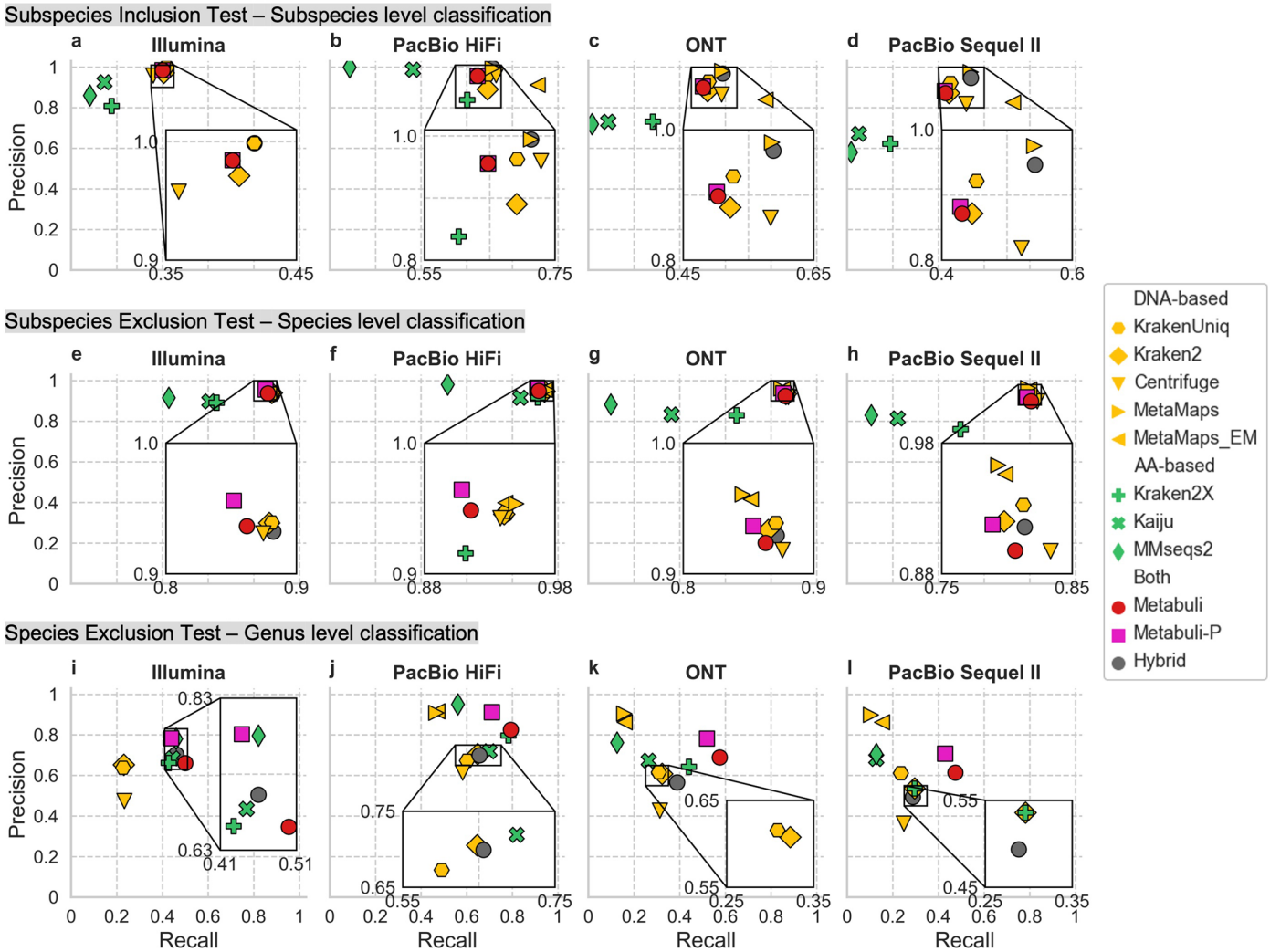
Extended data is available for this paper at https://doi.org/10.1038/s41592-024-02273-y.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-024-02273-y.

Correspondence and requests for materials should be addressed to Martin Steinegger.

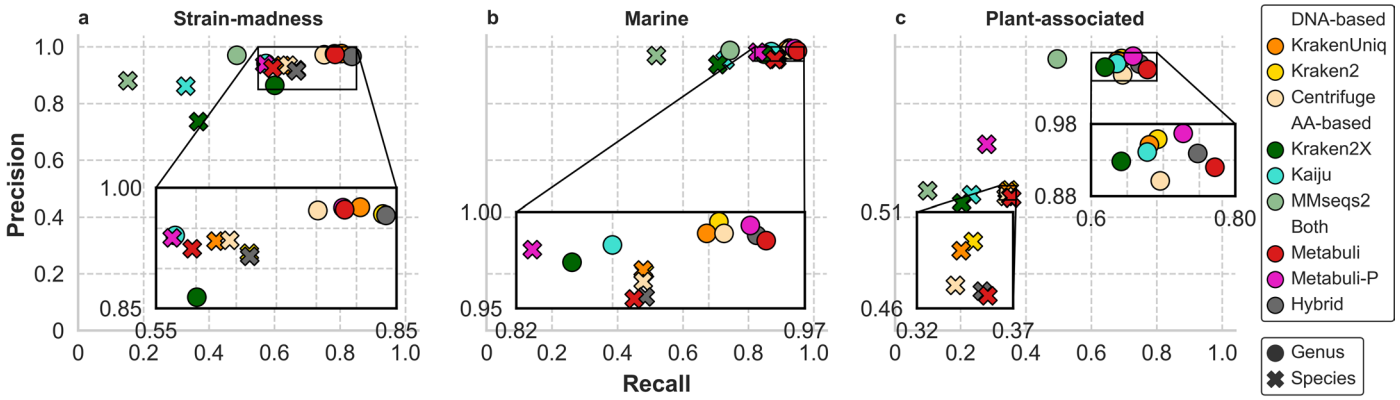
Peer review information *Nature Methods* thanks André Soares and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editor: Lei Tang, in collaboration with the *Nature Methods* team. Peer reviewer reports are available.

https://doi.org/10.1038/s41592-024-02273-y



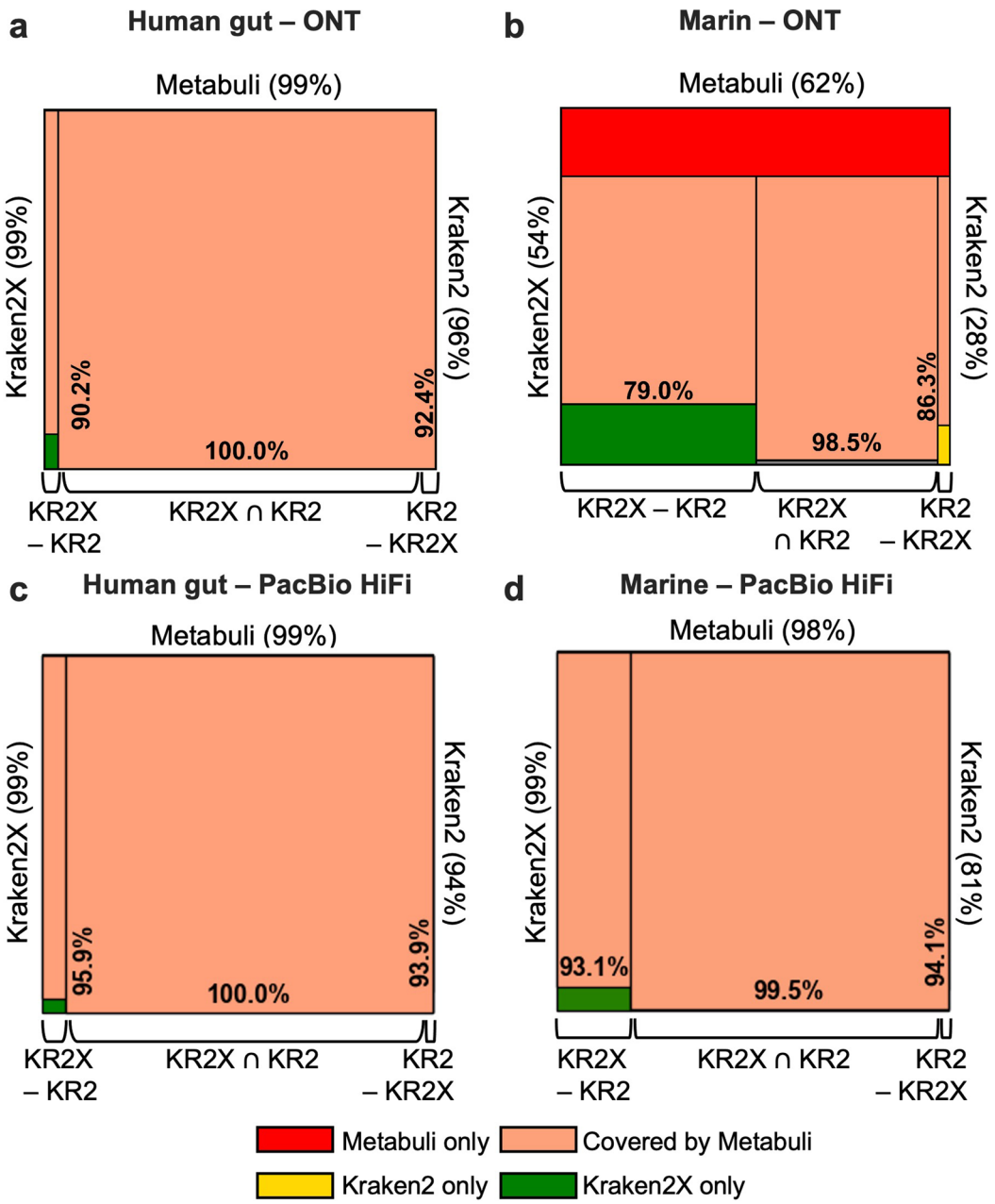
Extended Data Fig. 1 | Synthetic benchmark results. Simulated short (Illumina) and long (PacBio HiFi, ONT, and PacBio Sequel II) reads were used for performance evaluation based on GTDB genomes and taxonomy. Hybrid = (x, y) is the result of applying the DNA-based tool x, followed by the AA-based tool y, where both are the best-performing. **a**–**d** Subspecies-level classification tests. Reads were simulated from subspecies present in databases, and precision and recall were measured at subspecies rank. a) Hybrid = (KrakenUniq, Kraken2x). b-d) Hybrid = (MetaMaps, Kraken2X). Raw data for performance measurements at subspecies, species, genus, and family ranks are available in Supplementary Table 1. **e**–**h** Species-level classification tests. Not the queried subspecies but their sibling subspecies were contained in databases to measure species-level classification. Hybrid = (KrakenUniq, Kraken2X). Raw data for performance measurements at species, genus, family, and order ranks are available in Supplementary Table 2. **i**–**l** Genus-level classification tests. Not the queried species but their sibling species were contained in databases, so how well each tool can detect homology within the same genus was measured. i) Hybrid = (Kraken2, MMseqs2). j-l) Hybrid = (Kraken2, Kraken2X). Raw data for performance measurements at genus, family, order, and class ranks are available in Supplementary Table 3.

https://doi.org/10.1038/s41592-024-02273-y



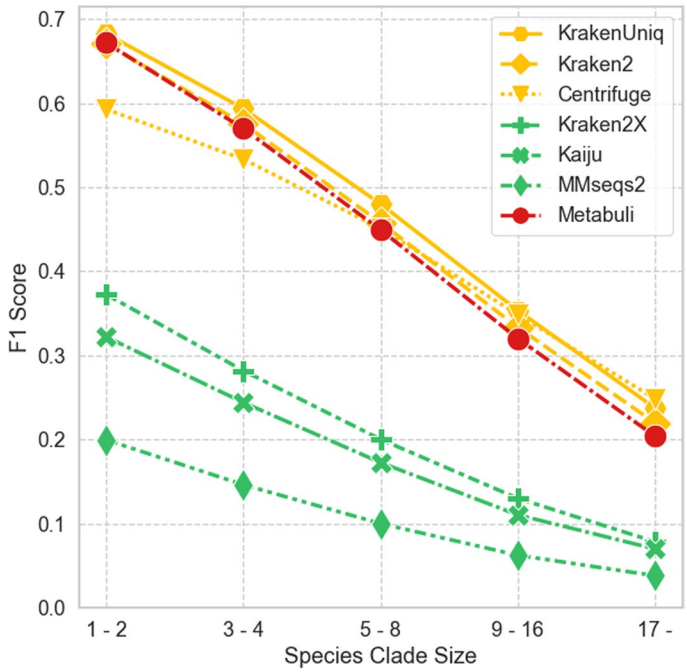
Extended Data Fig. 2 | Benchmarks using CAMI2's strain-madness, marine, and plant-associated datasets. GTDB genomes and the CAMI2-provided taxonomy were used for the database creation. CAMI2-provided short reads of strain-madness (**a**), marine (**b**), and plant-associated (**c**) datasets were classified by each tool, and the average values of the metrics that were measured at the species and genus rank for each sample were plotted. Raw data and metrics for each sample are available in Supplementary Tables 7–9.

https://doi.org/10.1038/s41592-024-02273-y



Extended Data Fig. 3 | Comparison of Metabuli to best performing AA- and DNA-based tools on real long-read metagenomic samples. In contrast to Fig. 2g–h, Kraken2X instead of Kaiju is utilized due to its superior performance on long reads. The databases were built using GTDB genomes and a human genome (T2T-CHM13v2.0) based on GTDB taxonomy edited to include a human taxon. Real nanopore sequencing data from human gut (**a**) and marine (**b**) environments, as well as PacBio HiFi reads from human gut (**c**) and marine (**d**) environments, were classified by each tool. The area is proportional to the number of reads within each panel. The proportion of reads classified by each tool is denoted in parentheses.

https://doi.org/10.1038/s41592-024-02273-y



Extended Data Fig. 4 | Subspecies-level classification performance by clade size. All 2,382 query subspecies used in Extended Data Fig. 1a were divided into groups according to the number of subspecies siblings they had in the reference database, that is, by their species clade size. The average F1 score for queries in each group decreases as the clade's size increases, indicating that more sibling subspecies pose a harder classification challenge to all tools. Precision and recall are available in Supplementary Table 10.

 https://doi.org/10.1038/s41592-024-02273-y

Extended Data Table 1 | Resource measurements in subspecies inclusion test

	Database		Illumina		PacBio HiFi	
Software	Size (GiB)	Time (min)	RAM (GiB)	Time (sec)	RAM (GiB)	Time (sec)
Kraken2	43	454	44	24	44	76
KrakenUniq	309	395	306	169	306	196
Centrifuge	40	736	41	218	41	249
Kraken2X	11	337	12	26	11	60
Kaiju	39	36	41	145	41	180
MMseqs2	37	6	174	6075	201	8106
MetaMaps	569	408	-	-	133	31 h 12 min
Metabuli DB	69	104	-	-	-	-
Metabuli 32 GiB	-	-	29	589	32	1296
Metabuli 64 GiB	-	-	55	555	62	1132
Metabuli 128 GiB	-	-	109	537	117	1158
Metabuli 256 GiB	-	-	173	546	228	1152
Metabuli 128 GiB 64t	-	-	109	400	120	838
Metabuli MacBook 8 GiB	-	-	5	6120	4	14671

* About 15 million 150 nt paired-end Illumina reads and 875,545 PacBio HiFi reads with an average length of 8958 nt were used across all simulations.
* Metabuli has an `--max-ram` option that limits RAM usage during classification. We measured its classification performance, setting this option to 8, 32, 64, 128, or 256 GiB.
* All runs utilized 32 threads except for "Metabuli Macbook" and "Metabuli 128GiB 64t", which used 8 and 64 threads, respectively.
* For Kraken2 and Kraken2X, the measured database creation time includes the optional step of masking low-complexity regions, which is not a necessary step but recommended in the software manual. This step took 254 and 153 minutes for Kraken2 and Kraken2X, respectively.
* For MetaMaps, classification time includes the time taken for its EM process.

N/A.

nature portfolio

Corresponding author(s): Martin Steinegger

Last updated by author(s): Mar 14, 2024

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed
☐	☒ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒ The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☐	☒ A description of all covariates tested
☐	☒ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☐	☒ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☐	☒ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	Benchmarked software. (Please see "Benchmarked software" section in Methods for software description) - Metabuli (v1.0.2) https://github.com/steineggerlab/Metabuli - KrakenUniq (v0.7.3) https://github.com/fbreitwieser/krakenuniq - Kraken2 (v2.1.2) https://github.com/DerrickWood/kraken2 - Kaiju (v1.9) https://github.com/bioinformatics-centre/kaiju - Centrifuge (v1.0.4) https://github.com/DaehwanKimLab/centrifuge - MMseqs2 Taxonomy (release 13.45111) https://github.com/soedinglab/mmseqs2 - Metamaps (v0.1) https://github.com/DiltheyLab/MetaMaps Software used to simulate short and long reads. - Mason2 (v2.0.9) https://github.com/seqan/mason - PBSIM3 (v3.0.0) https://github.com/yukiteruono/pbsim3 Software used to download GTDB genomes and predict genes in them. - Struo (v0.1.7) https://github.com/leylabmpi/Struo - Prodigal (v2.6) https://github.com/hyattpd/Prodigal
Data analysis	Scripts used for running benchmarks and drawing plots are deposited in "https://github.com/jaebeom-kim/metabuli-analysis" and "https://github.com/jaebeom-kim/metabuli-plots", respectively.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

The raw data for Fig. 2 and Extended Data Figure 1-4 are provided as Source Data. Performance measures at different ranks of benchmarks of Fig. 2a-b and Extended Data Fig. 1 are available in Supplementary Table 1-3. The assemblies used for read simulation and database creation in synthetic benchmarks are listed in Supplementary Table 4, and the simulated reads are available at https://doi.org/10.5281/zenodo.10250585. More detailed results and utilized accessions of Fig. 2c-d are provided in Supplementary Tables 5-6. The databases used in Fig. 2c-d were built using viral genomes (release 212) and a human genome (GCF_009914755.1) downloaded from NCBI RefSeq, and accessions of genomes of analyzed SARS-CoV-2 variants were denoted in "Pathogen detection tests" section in Methods. Performance measures at different ranks of Fig. 2e and Extended Data Fig. 2 are provided in Supplementary Table 7-9. Precision and recall of Extended Data Fig.4 are available in Supplementary Table 10. The accessions of real data analyzed in Fig. 2g-h and Extended Data Fig. 3 are denoted in "Benchmarks with real metagenomes" section in Methods. CAMI2-provided datasets and taxonomy used in Fig. 2e-f and Extended Data Fig. 2 can be downloaded from https://data.cami-challenge.org/participate.

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	Not applicable.
Population characteristics	Not applicable.
Recruitment	Not applicable.
Ethics oversight	Not applicable.

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	When we determine the genomes for database creation and query read simulation, we used as many as genomes that meet the criteria for benchmark design. The criteria are described in Methods sections for each benchmarks.
Data exclusions	CAMI2 provides 10, 21, and 100 query samples for the marine, plant-associated, and strain-madness benchmarks, respectively. To reduce the run-time of the benchmarks, we took all, every second, and every tenth query samples, respectively.
Replication	To demonstrate the specificity-sensitivity trade off, we compared DNA-based tools, AA-based tools, and Metabuli with simulated short and long reads, real RNA-seq reads from COVID-19 studies. The results of the comparison confirmed the trade off well.
Randomization	When we choosing genomes from certain taxa, we randomly choose them. For example, when we estimated genus-level sequence similarity score, we randomly choose genomes of two species (one for DB and the other for query) from genus that have at least two species.
Blinding	We didn't divided query samples to multiple groups.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging