# Benchmarking the impact of reference genome selection on taxonomic profiling accuracy

**Jasper van Bemmelen[1], Ioanna Nika[1], and Jasmijn A. Baaijens[1,2,*]**

[1]Intelligent Systems Department, Delft University of Technology, Delft, The Netherlands
[2]Department of Biomedical Informatics, Harvard Medical School, Boston, USA

*Correspondence address. Jasmijn Baaijens, Intelligent Systems Department, Delft University of Technology, Van Mourik Broekmanweg 6, 2628 XE Delft, The Netherlands. E-mail: j.a.baaijens@tudelft.nl

**Abstract**

**Background**: Over the past decades, genome databases have expanded exponentially, often incorporating highly similar genomes at the same taxonomic level. This redundancy can hinder taxonomic classification, leading to difficulties distinguishing between closely related sequences and increasing computational demands. While some novel taxonomic classification tools address this redundancy by selecting a subset of genomes as references, insights regarding the impact of different reference genome selection methods across taxonomic classification tools are lacking.
**Results**: We systematically evaluate genome selection and dereplication methods on bacterial and viral datasets using simulated metagenomic samples. We show that the impact of reference genome selection is strongly context-dependent. For bacterial profiling, incorporating all available genomes generally yields the highest accuracy, while having a limited impact on computational resource usage. In contrast, for highly redundant SARS-CoV-2 datasets we find that stringent hierarchical clustering-based selection significantly improves lineage-level abundance estimation accuracy. Incorporation of location-based metadata further enhances viral profiling performance by prioritizing locally relevant genomes. Across viral experiments, smaller reference sets significantly reduce memory and runtime requirements during both indexing and profiling, although this comes at an additional pre-processing cost.
**Conclusions**: Reference genome selection influences both accuracy and computational efficiency in taxonomic profiling, although its benefits seem context-dependent. In diverse bacterial communities, comprehensive reference sets appear optimal, whereas in redundant viral datasets smaller and metadata-informed reference sets work best. These results demonstrate that reference set design does not have a one-size-fits-all solution, and that selection strategies should be adapted based on the biological and computational setting.

**Keywords:** Taxonomic profiling; Metagenomics; Sequence dereplication; Sequence clustering; Genome selection

## 1 Introduction

Taxonomic profiling forms one of the cornerstones of metagenomic analyses, allowing researchers to characterize and contextualize metagenomic samples by comparing samples against a database with a known taxonomy. Over the last decades, we have seen an explosion in the availability of genomic data, with databases hosted by the National Center for Biotechnology Information (NCBI) [1] and the Genome Taxonomy Database (GTDB) [2] doubling in size every few years. As taxonomic profiling inherently relies on the use of genome databases, this means that taxonomic profiling tools have to cope with their explosive growth.

In recent years, many new taxonomic profiling tools and pipelines have been developed. Several of these explicitly address the consequences of ever-increasing genome databases. For example, Kraken2 [3], one of the most popular taxonomic profiling tools, efficiently deals with large taxonomic databases by representing genomes as minimizers (a subset of the constituent $k$-mers of a sequence). This significantly reduces the storage and memory requirements compared to its predecessor. In a similar fashion, CLARK [4] only stores $k$-mers that are unique to the genomes in the database. Instead of using a compressed $k$-mer-based representation, Centrifuge [5] progressively builds a pangenome for every species by adding non-redundant segments of genomes. Marker-based taxonomic profiling tools, like MetaPhlan [6] and mOTUs [7], use databases which only store marker gene sequences. All of the profiling tools above focus on compressed representations of a database, rather than sequence selection.

Taxonomic profiling can also benefit from excluding redundant or less informative sequences through careful selection of representative genomes. Besides improving scalability, this approach can enhance profiling accuracy.

Genome databases have taken a first step by selecting curated, high-quality genome sequences. A primary example is NCBI's RefSeq [8], one of the most widely used genome databases, which has a selection of representative genomes for over 28,000 named species over all taxonomic kingdoms [9] and serves as a standard for many taxonomic profiling tools. Similarly, GTDB, which currently hosts over 600,000 microbial genomes, also provides a set of high-quality (i.e., low contamination, high completeness) reference genomes. Its representative genomes define species boundaries within the taxonomy and are updated as higher-quality genomes become available [2]. These curated databases play an important role in enabling accurate and efficient taxonomic profiling.

However, our understanding of optimal strategies for genome selection in the context of taxonomic profiling is limited and further selection may be desired. Recent methods [10, 11] incorporate a selection step to filter out highly similar genomes during index generation, which reduces false positive rates [10]. This step, often called sequence dereplication, selects a small subset of sequences such that every sequence is represented by at least one selected sequence. Tools and software suites that do so include CD-HIT [12], UCLUST [13], MeShClust [14] and MMseqs2 [15], and were primarliy developed to dereplicate many short sequences such as reads. Nevertheless, it remains unclear how widely applicable such selection strategies are and to what extent they can improve taxonomic profiling pipelines through reference genome selection.

While several benchmarking studies have compared the performance of different taxonomic profiling tools ([16, 17, 18, 19, 20]), these rely on a fixed selection of reference genomes to ensure fair comparisons across methods. Some studies have explored the effects of different reference databases, but only for different versions of RefSeq [21] or other standard databases in combination with different profiling parameters [22, 23]. Other work has focused on the use of taxon-specific reference sets, highlighting how this can bias results [24], or how dereplication tools scale in the context of read or otherwise short sequence dereplication [25, 26, 27]. However, to the best of our knowledge, no studies have systematically examined how variations in reference selection—beyond database versions or taxon-specific choices—affect the performance of taxonomic profiling tools.

To address this gap, we assess the impact of different genome selection techniques as a pre-processing step in taxonomic profiling. We consider a global database of potential reference genomes and apply a range of sequence dereplication methods, including VSEARCH [28], MeShClust [14], Gclust [29], and GGRaSP [30], as well as hierarchical clustering approaches. These methods aim to filter out sequences that are highly similar to other sequences thereby framing the problem of reference genome selection as a specific instance of sequence dereplication. Using the filtered reference sets generated by these methods, we construct taxonomic profiling indexes and benchmark their effect on profiling accuracy in simulated metagenomic samples. We find that, when target genomes are highly similar, using hierarchical clustering to select reference genomes can significantly improve taxonomic profiling accuracy while reducing computational resource requirements. When target genomes are less similar, we instead find that more sophisticated dereplication methods like MeShClust with a high similarity threshold can improve accuracy. This highlights the value of reference genome selection in optimizing taxonomic profiling.

## 2 Overview of sequence dereplication methods

Sequence dereplication tools aim to reduce redundancy in a set of sequences by finding a subset of representative sequences. To do so, they address the following coverage-based problem: given a set of sequences $S$, find a subset $S' \subseteq S$ that is representative of $S$. Although definitions of representative can depend on the dereplication method, the most commonly considered definition defines a subset $S' \subseteq S$ to be representative of $S$ if, for a given similarity function $f$ and similarity threshold $T$, it holds that for every $s \in S$, there exists at least one sequence $s' \in S'$ such that $f(s, s') \geq T$. The key idea in this formulation is that every sequence in $S$ is represented by some sequence in $S'$. Below, we describe several categories of existing dereplication methods.

### Greedy incremental clustering

Greedy incremental clustering methods have been developed to operate on large numbers of sequences, making them applicable to modern day sequence volumes. Since comparing sequences based on exact alignment is computationally expensive [31], and the number of comparisons scales quadratically with the number of sequences, these tools focus on minimizing the number of sequence comparisons. In the greedy incremental clustering paradigm, sequences are sorted according to a total order (often from longest to shortest) and the first sequence is selected as a representative of its own cluster. Afterwards, subsequent sequences are compared to existing representatives, adding them to an existing cluster if their similarity with the corresponding representative is larger than the given threshold $T$. If no such sequence exists among the currently selected representatives, the

sequence instead becomes a representative of its own cluster.

Many popular sequence dereplication tools, like CD-HIT [12], UCLUST [13], VSEARCH [28] and Gclust [29] utilize the greedy incremental clustering framework. These methods have distinct implementations which differ primarily in how they define the similarity function $f$, or in how they apply filters to further reduce the number of comparisons that need to be made. For example, CD-HIT performs global alignment and then defines the similarity as the number of identical characters divided by the length of the shortest sequence of the two, whereas UCLUST divides by the average length of both sequences.

### Submodular optimization

In addition to the greedy incremental clustering paradigm, there are also methods that consider a coverage-based problem formulation, but instead solve it differently. In Repset [32], it is shown that problems adhering to a coverage-based selection formulation are related to unconstrained problems in which a submodular function is optimized instead. By using known approximation algorithms for optimizing submodular functions, [32] shows that this dereplication works well for dereplicating protein sequences. Furthermore, they also show that extensions can be incorporated with allows for bi-objective functions in which the number of selected representatives is explicitly minimized. However, the authors also show that, compared to greedy incremental clustering methods, their approach is significantly slower, making it less applicable to large scale datasets with millions of sequences.

### Mean-shift clustering

Mean-shift clustering provides an alternative approach where cluster representatives are iteratively shifted towards the center of mass of their respective clusters. MeShClust [14] is based on a mean-shift [33] algorithm where pairs of clusters are merged when the pairwise similarity of their representatives exceeds a similarity threshold. Representatives are re-selected every iteration as the sequence closest to the center of their respective clusters. Therefore, this approach finds more meaningful clusters and cluster representatives compared to approaches that fix representatives, by allowing cluster representatives to be updated [34]. Additionally, MeShClust incorporates a generalized linear model that is trained to classify pairs of sequences as being similar according to the provided threshold $T$. While this requires training a model upfront, it results in relatively fast sequence comparisons during the mean-shift algorithm. Hence, MeShClust can run on thousands of full-length bacterial genomes ($\sim$3Mbp) despite the computational complexity of the mean-shift algorithm [14].

### Hierarchical clustering

Another popular strategy for sequence dereplication is to apply hierarchical clustering, after which representatives per cluster are selected. This approach is implemented by both dRep [35] and GGRaSP [30], and is also already used as a pre-processing step by recent taxonomic profilers such as StrainGE [11] and YACHT [10]. In order to determine appropriate clusters, GGRaSP either allows for a user-defined similarity threshold $T$, or alternatively uses Gaussian mixture models to find a suitable threshold. After clustering, GGRaSP returns the medoid sequence of every cluster as a representative, but also allows for a ranking of the input sequences to prioritize the selection of sequences that meet certain user requirements (e.g. assembly completeness).

One of the downsides of GGRaSP is that it requires the user to input the relationship between input sequences by either providing a phylogeny, an all-pairs distance matrix or a multiple alignment. In contrast, dRep allows for estimating sequence similarities directly by using fast approximation techniques like MASH [36]. Moreover, instead of a single round of hierarchical clustering, dRep operates in two steps: first a rough clustering of the input sequences is obtained using a more lenient similarity threshold than $T$. Afterwards, dRep performs hierarchical clustering again within every cluster using the specified threshold $T$ and a more precise similarity estimator like fastANI [37]. Because of the initial rough clustering, unnecessary pairwise similarity estimations with the precise estimator can be skipped. Finally, representative genomes are selected from the resulting fine-grained clusters by scoring sequences according to quality metrics like completeness and contamination for genomes.

While both methods were intended to work fast in practice, they suffer from the fact that they require all-vs-all similarity calculations. Although this can be alleviated through the use of fast similarity estimators like MASH and fastANI, this still means that every pair of sequences has to be compared at least once. Consequently, methods that utilize hierarchical clustering for sequence dereplication will have difficulty scaling to millions of sequences.

### MMseqs2

The MMseqs2 [15] software suite contains several methods for sequence dereplication that were built for large scale protein sequence dereplication. Most of these methods consider the dereplication problem as a graph-based problem, in which connected components or otherwise connected subgraphs are iteratively removed until the graph is empty. In doing so, the subgraphs that are removed are deemed a cluster, and a representative sequence per cluster is selected such that every sequence in the original set $S$ is represented by some selected sequence. Additionaly, MMseqs2 also contains a clustering module that performs a hierarchical clustering-based selection similar to dRep and GGRaSP, but selecting the longest sequence in every cluster as a representative by default.

In addition to the graph-based clustering methods, MMseqs2 also includes a clustering algorithm called LINCLUST [38] that was developed to cluster sequences in near-linear time. To achieve this, LINCLUST utilizes a minimizer-based criterion that defines groups of sequences sharing at least one $k$-mer. Afterwards, the longest sequence of every group is chosen as its center, and groups that share a center are merged. Finally, sequences are aligned to the center of their groups, and sequences are uniquely assigned to a group based on alignment derived metrics, retaining only the center of every group. In their paper, the authors show that LINCLUST's runtime scales linearly with the number of sequences in practice, making it feasible to scale to billions of sequences.

### Alternative dereplication methods

In addition to the methods listed above, there are also methods that have been developed to dereplicate sequences under specific conditions. PARNAS [39], for example, treats the dereplication problem as a $k$-medoids problem or optimal coverage problem on a phylogenetic tree. By requiring a phylogenetic tree structure, distances between sequences adhere to a tree-metric, which makes the corresponding dereplication problems computationally feasible. Another method for dereplication is TARDiS [40], which uses a genetic algorithm to select a pre-defined number of sequences while optimizing for maximal sequence diversity, as well as time distribution. This allows for the incorporation of temporal metadata of the sequences if this is available, but relies on determining the number of sequences to find beforehand.

In contrast to explicitly selecting representative sequences through dereplication, ReprDB and panDB [41] were introduced as a method to find a compressed representation of the pangenome of a species. These methods work by starting with an initial representative genome, followed by iteratively aligning genomes from the same taxonomic group (i.e. target) in order to detect novel genomic fragments that are added to the pangenome representation, similar to the approach used by Centrifuge. Although this method allows for a compressed representation of a set of genomes, it does not select representatives genomes directly.

## 3 Methods

The general reference set construction workflow involves three steps: genome retrieval, optional quality filtering or downsampling, and sequence dereplication (Figure 1). This section provides an overview of the tools used, followed by a description of datasets, benchmarking setup, and performance evaluation.

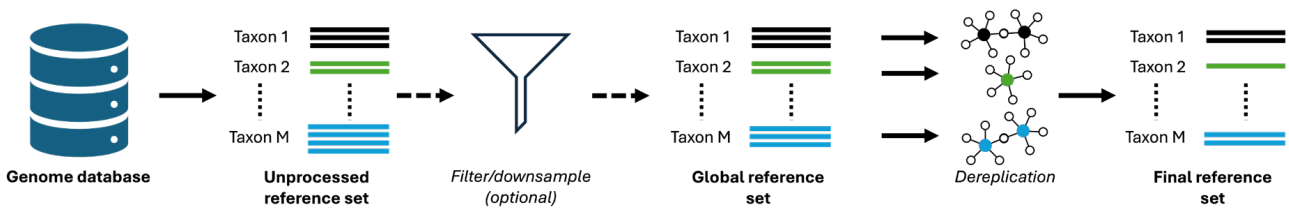


**Figure 1.** A schematic representation of the reference genome selection pipeline. Potential reference genomes are first retrieved from a database. Afterwards, quality filtering and/or downsampling can be applied. Finally, sequence dereplication methods are applied to the genomes of every taxon of interest (individually), resulting in a final reference set.

### 3.1 Selection of dereplication tools

For our experiments, we made a selection of sequence dereplication tools representing the major methodological categories outlined in the previous section.

From the greedy incremental clustering family, we included VSEARCH and Gclust. Unlike VSEARCH, UCLUST was omitted since the unrestricted 64-bit version is only available as commercial software while having similar functional capabilities as VSEARCH. Gclust, which uses sparse suffix arrays in order to accelerate sequence similarity searches in the incremental clustering framework, was included as an efficient alternative to VSEARCH. CD-HIT was initially tested, but failed to generate results even for smaller subsets of the data due to time limitations and was therefore excluded. Similarly, the submodular optimization-based dereplication tool Repset was also excluded due to efficiency considerations

As the sole representative of the mean-shift clustering approaches, we included MeShClust in our experiments. Between the available versions (v1 and v2/3) we worked with the latest available release since it has been shown to produce higher quality clusters than its predecessor [14].

Further, we included two hierarchical clustering-based approaches. The first, GGRaSP, was included as it incorporates a novel method for automatically determining similarity thresholds using Gaussian mixture models. The second is a custom hierarchical clustering implementation utilizing the SciPy library (https://scipy.org). This allows us to reuse the similarity matrix generated for GGRaSP whereas dRep computes similarities at runtime. Moreover, in generating a pool of reference genomes we already filter based on quality metrics similar to those used by dRep, making its built-in quality metric scoring largely redundant.

For the viral experiments we also included the Viral Lineage Quantification (VLQ) selection pipeline [42] as an additional reference genome selection strategy. This method was specifically developed for SARS-CoV-2 genome dereplication and greedily selects sequences that have mutations (relative to a universal reference genome) that occur with a frequency of at least 50%.

Tools from the MMseqs2 suite (including LINCLUST) were excluded in our experiments since they failed to produce results while setting up experiments, similar to CD-HIT. Although the cause is different (CD-HIT failed to complete within five days, MMseqs2 had runtime errors), this has led us to exclude all of the corresponding implementations. Finally, PARNAS was excluded since it requires the input to adhere to a tree-metric, and TARDiS was excluded as we do not incorporate temporal metadata explicitly in our experiments.

### 3.2 Datasets

We designed our experiments to consider two main scenarios: bacterial genomes and viral genomes (specifically SARS-CoV-2). For the bacterial scenario, the focus is on taxonomic classification at the species level, whereas for the viral scenario, the goal is estimate abundances at the lineage level. This setup allows us to investigate how the impact of reference genome selection differs between datasets with varying degrees of sequence similarity. We focus on impact in terms of taxonomic profiling accuracy, but we also report the efficiency of the dereplication, index construction, and taxonomic profiling in relation to the different dereplication tools used.

Given the differences in resolution required for these scenarios, we used different profiling methods. For the bacterial scenario we use Bracken [43], Centrifuge [5] and DUDes [44]. For the viral scenario, we use the VLQ pipeline. We describe the datasets for each scenario in more detail below.

*Bacterial experiments*

For our bacterial experiments we performed three sets of experiments based on NCBI's taxonomy (downloaded September 24th, 2024): i) bacterial species from the same order, ii) bacterial species from the same family, and iii) bacterial species from the same genus. The motivation for this set-up is that it includes taxonomic groups with different levels of relatedness, allowing us to examine how the impact of reference genome selection may vary with the overall similarity of the target genomes. These settings will be referred to as "order experiments", "family experiments", and "genus experiments", respectively. In each experiment, we performed sequence dereplication at the species level (Figure 1), using MeShClust (Identity v1.2, MeShClust v2.0), Gclust (v1.0), GGRaSP (v1.3), and a custom implementation of hierarchical clustering-based selection. Afterwards, we performed taxonomic profiling at the species-level of simulated metagenomic samples to determine the accuracy achieved by every sequence dereplication strategy.

**Table 1.** Experimental setup: selected genus, family and order taxa, including the corresponding species selected for every experiment.

| Experiment | Selected taxon | Selected species |
|---|---|---|
| Genus | *Streptococcus* | *Streptococcus suis*<br>*Streptococcus thermophilus*<br>*Streptococcus agalactiae*<br>*Streptococcus pneumoniae*<br>*Streptococcus pyogenes* |
| Family | *Enterobacteriaceae* | *Citrobacter freundii*<br>*Escherichia coli*<br>*Salmonella enterica*<br>*Enterobacter hormaechei*<br>*Klebsiella quasipneumoniae* |
| Order | Enterobacterales | *Buchnera aphidicola*<br>*Citrobacter freundii*<br>*Escherichia coli*<br>*Serratia marcescens*<br>*Enterobacter hormaechei* |

This table summarizes the taxa and species chosen for each bacterial experimental setup.

**Dataset composition** The genomes used throughout our experiments consisted of all "Complete Genome" status assemblies from NCBI Assembly (now NCBI Genomes), downloaded on October 9th, 2024. In this, we only included species that had at least one such assembly.

The selected order, family and genus are shown in Table 1 and were obtained as follows: first we filtered out any species that had fewer than 100 genomes in the database, as our primary interest lies in dereplicating genomic sequences. Afterwards, remaining species were sorted according to the average intra-species distance (estimated using MASH as explained below), from highest to lowest. This prioritizes species that are relatively diverse and are expected to have the largest impact when dereplicated. For all experimental settings, the selected taxon was the first taxon to which 5 species belonged, by parsing the sorted species list in order. Although we selected 5 species from the selected taxa, all other species of the selected taxa with at least one "Complete Genome" quality assembly were also included when running the dereplication and taxonomic profiling tools.

**Parameter settings.** All tools were run with similarity thresholds of 95%, 97% and 99%, except for GGRaSP for which we used the Gaussian mixture model based threshold selection strategy. The 95% similarity threshold reflects the cut-off for species definitions in prokaryotes [45], and the higher thresholds (97%, 99%) may create more refined reference sets. The exact commands (including parameter settings) used to run every tool can be found on Github [46], and the number of threads used for every method are given in Supplementary Tables S1-3. For both GGRaSP and the other hierarchical clustering-based selections we used MASH v2.3 with the default $k$-mer size of 21 and a sketch size of 1000 to estimate distances between all pairs of genomes within a species. For the hierarchical clustering selections we used scikit-learn v1.5.0 [47], and generated selections using both single-linkage clustering and complete-linkage clustering at the designated thresholds. Representatives for each cluster were obtained by selecting the centroid of respective clusters. We generated a centroid reference set containing only the centroid genome of every species as a naive selection strategy. If a method failed to produce a selection for a species, we selected a random reference genome to represent the species. Finally, as a baseline, we include the "All" reference set, which contains all genomes (no selection).

**Index construction and taxonomic profiling** For profiling we use Bracken v1.0.0 [43] (building on Kraken2 v2.1.3 [3]), Centrifuge v1.0.4.2 [5], and DUDes v.0.10.0 [44] (using BWA v0.7.18 [48] for alignment). This selection of methods covers two popular $k$-mer based profiling tools with different approaches for compressing reference genomes (Bracken, Centrifuge) as well as one exact alignment based tool (DUDes). For all methods, reference indexes were constructed using default parameters, except for the number of threads (Supplementary Tables S1-3).

One complicating factor is that these taxonomic profiling tools do not estimate taxon abundances in the same way. Bracken estimates "sequence abundance", which is defined as the number of reads assigned to a particular

taxon divided by the total number of reads that have been classified. In contrast, DUDes estimates "taxonomic abundance", which is defined as the relative number of occurrences of a particular taxon in a sample. The core difference between these abundance definitions is that taxonomic abundances account for genome lengths and thus these outcomes cannot be compared directly [49]. Centrifuge estimates both sequence abundance as well as taxonomic abundance, but we evaluate the sequence abundance.

**Read simulations** To assess the performance of taxonomic profilers using different reference sets, we simulated paired-end Illumina reads for every experiment using ART v2016.06.05 [50]. For all bacterial experiments, we generated reads with a length of 150 basepairs (bp), and a mean fragment size of 270bp with a standard deviation of 20bp. Furthermore, for every experiment we simulated 10 metagenomic samples with approximately 4,500,000 HiSeq2500 read-pairs generated from 10 genome assemblies from NCBI Assembly. Genome assemblies included for read simulations consisted of the top 10 assemblies, based on completeness, and were required to have a "Scaffold" or "Chromosome" assembly quality. Throughout all experiments, the sequence abundance of every species was fixed at approximately 20% (uniformly distributed over the 10 genomes for every species), with taxonomic abundances varying based on genome lengths.

*Viral experiments*

For the viral experiments, we focused on SARS-CoV-2 to evaluate the effect of reference genome selection on lineage-level abundance estimation. We created simulated samples representing SARS-CoV-2 populations detected in wastewater at a specific point in time in Connecticut, USA. Similar to the bacterial experiments, we performed 3 sets of experiments, differing in the geographic filtering applied to the genome database (GISAID [51]): i) global experiments, with no geographic filter; ii) country experiments, including only reference genomes from the same country (USA) as the samples; and iii) state experiments, including only genomes from the same state (Connecticut) as the samples. This setup allows us to explore whether incorporating geographic information into reference genome selection can improve lineage-level abundance estimation, an aspect particularly relevant for environmental samples such was wastewater. In each experiment, we performed sequence dereplication at the lineage level (Figure 1. In addition to the dereplication methods used in the bacterial experiments, we included VLQ's allele frequency-based selection method [42] and VSEARCH v2.28.1 (which was infeasible on bacterial genomes due to their length). We used the VLQ pipeline [42] to perform lineage-level abundance estimation and evaluate abundance estimation accuracy similar to the bacterial experiments.

**Dataset composition** The genomes used throughout the viral experiments were downloaded from GISAID [51] on June 12, 2022, and consisted of sequences from the period January 1st 2021 until March 31st 2021, satisfying a genome length of at least 25,000bp, and an N-content of at most 0.1%. After geography-based selection, genomes were randomly downsampled to at most 1000 sequences per lineage to keep computations feasible.

**Parameter settings** For both GGRaSP and the other hierarchical clustering methods, we used MASH to estimate pairwise sequence similarities, using a sketch size of 5000 and $K$-mer size of 31. We ran GGRaSP with the automatic threshold selection strategy. For the single-linkage and hierarchical-linkage clustering selections we calculated the 1st, 5th, 10th, 25th, 50th, 90th and 99th percentile distances for every lineage individually, and used these as similarity thresholds. VSEARCH, Gclust and MeShClust define similarities during selection, so for these tools we used pre-defined similarity thresholds of 95%, 99%, and 99.9% (SARS-CoV-2 genomes are generally more than 99% similar). For MeShClust, the maximum allowed similarity threshold is 99%, so we used only the 95% and 99% thresholds.

For the selection with VLQ we used a minimal allele frequency threshold of 50%, which ensures that given considered genomes, all point-wise mutations (compared to the MN908947.3 reference genome) that occur with a frequency of at least 50% (within their respective lineage) are captured in the selected genomes. We select a random genome for lineages where a method failed to successfully select genomes. Finally, we consider centroid and "All" reference sets, similar to the bacterial setting.

**Index construction and taxonomic profiling** Following the VLQ pipeline, we construct reference indexes using kallisto v0.44.0 [52]. Both index construction and subsequent abundance estimation were done using default parameters, and abundance estimations were further processed using the scripts available in the VLQ pipeline that convert kallisto output (transcripts per million) to relative abundances at the lineage level.

**Read simulations** For the viral experiments we simulated paired-end Illumina HiSeq2500 reads now using SARS-CoV-2 genomes having Connecticut as their marked location, and April 30th, 2021 as their sampling date. This constitutes a collection of 25 genomes used for simulations, spread across 6 lineages. We simulated 20 samples of approximately 9,000-10,000 paired-end reads with a length of 150bp, and a mean fragment size and standard deviation of 250bp and 10bp respectively. Since preliminary experiments showed large effects induced by reference genome selection, we repeat these experiments fixing the relative abundance of lineage B.1.1.7 (commonly known as the alpha variant) at 1%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100%. Background sequences were randomly simulated from other lineages, and this set-up allows us to assess whether the effect of reference genome selection on profiling accuracy holds at different sample compositions.

### 3.3 Performance evaluation

All experiments presented in this work were run on the Delft Artificial Intelligence Cluster (DAIC) high performance computing cluster [53]. The benchmarking pipeline and commands are publicly available on Github. In what follows we will describe how we compare experimental outcomes.

*Reference set comparisons*

In all experiments, we calculate containment indices (CI) between every pair of reference sets. Formally, the CI is defined as $\text{CI}(R_1, R_2) = \frac{|R_1 \cap R_2|}{|R_1|}$, where $R_1$ and $R_2$ are sets of reference genomes. For our results, we only consider genomes from species that have at least 2 sequences to calculate containment indices, since otherwise the intersection for a given species is always equal to 1. Similar to the Jaccard index, the containment index quantifies the similarity of sets. Unlike the Jaccard index, however, the containment index is asymmetric in its arguments, providing additional information when reference sets differ in size.

We also assess statistically whether the observed containment indices are greater than what could randomly be expected. We constructed an empirical null distribution using Monte Carlo simulations. For every taxonomic group $g$ (species for bacterial experiments, lineages for viral experiments), given the total number of genomes $N_g$ and counts $n_{1g}$, $n_{2g}$ in reference sets $R_1$ and $R_2$ respectively, we sample overlaps from a hypergeometric distribution to simulate random selection of $n_{1g}$ genomes given $n_{2g}$ genomes already chosen. Summing simulated overlaps across all taxonomic groups yields a total overlap per simulation. Afterwards, we compute the corresponding containment index for every simulated draw (10,000 per containment index) and compare against the observed containment index to obtain a one-sided p-value testing whether the observed similarity was larger than expected.

Since multiple containment indices were tested ($15 \times 15$ for bacterial experiments and $26 \times 26$ for viral experiments), we corrected for multiple hypothesis testing using the Benjamini-Hochberg procedure applied to all off-diagonal p-values excluding comparisons versus the "All" reference set. Adjusted p-values below 0.05 were considered statistically significant.

*Taxonomic profiling evaluation*

To evaluate the accuracy of taxonomic profiling we calculate both the L1-norm and F1-score of abundance estimates compared to the groundtruth for every profiler, sample and reference set. As in OPAL [54], we define the L1-norm of predicted abundances $\hat{\phi}_t$ for all taxa $t$ as:

$$\text{L1}(\{\hat{\phi}_{\mathbf{t}}\}) = \sum_t |\phi_t - \hat{\phi}_t| \tag{1}$$

where $\phi_t$ denotes the groundtruth abundance of taxon $t$. The L1-norm measures the total error made between the true and predicted abundances, and ranges from 0 to 2. We transform the computed L1-norms in the following way:

$$\overline{\text{L1}}(\{\hat{\phi}_{\mathbf{t}}\}) := 1 - \frac{L1(\{\hat{\phi}_{\mathbf{t}}\})}{2} \tag{2}$$

to obtain what we will denote the *abundance accuracy*. The abundance accuracy is equal to one minus the Bray-Curtis dissimilarity for abundance estimates and ranges from 0 to 1, where a value of 1 indicates a perfect estimation of all abundances, and a value of 0 indicates a totally incorrect estimation of abundances.

The F1-score is defined as the harmonic mean of the precision and recall of estimations, and ranges between 0 and 1. A value of 1 indicates that both the precision and recall are perfect, meaning that every taxon with nonzero abundance in the groundtruth has a nonzero estimated abundance, and every taxon with zero groundtruth abundance also has zero estimated abundance.

We reduce false positives by removing any taxa with predicted relative abundance below 0.1%, and subsequently re-scale remaining abundances to sum up to 100%. For all experiments, profiling methods, and reference sets, we compare both the abundance accuracy and F1-score to the respective scores of the "All" selection. We test whether a selection achieves significantly higher accuracy than the "All" selection (for both abundance accuracy and F1) by performing a Wilcoxon signed rank test, adjusting for false discovery rate using Benjamini-Hochberg correction within every experiment and for every profiler and metric separately. Adjusted p-values below 0.05 were considered statistically significant.

# 4 Results

## 4.1 Reference set compositions overlap across dereplication methods

To gain insight into reference set similarity across different dereplication tools, we calculated containment indices (see Methods) between reference sets. In calculating containment indices we ignored taxa (i.e. species or lineages) for which the database contained a single genome, as these are included in all reference sets by default.

Figure 2**(A)** shows that in the bacterial experiments, different reference sets have substantial overlaps with containment indices often exceeding 0.50. This effect is particularly prominent when comparing smaller reference sets (e.g. those selected through hierarchical clustering) to large reference sets, or when comparing different reference sets obtained from the same method using different similarity thresholds. Interestingly, the overlaps between all MeShClust and Gclust selections are high in general, despite their respective selection methodologies being largely distinct.

When comparing reference sets $R_1, R_2$ with $|R_1| \gg |R_2|$, the containment index is generally small: the numerator in the containment index is bounded by the smaller reference set, while the denominator is determined by the larger reference set. Thus, to assess whether the containment indices are larger than expected by chance, we compare to an empirical null distribution and test for statistical significance (see Methods). In Figure 2, containment indices which are significantly larger than expected are indicated by an asterisk. We observe that even for small containment indices, their similarity is generally significantly larger than expected by chance. We observe the same overall trends for the reference sets corresponding to the selection of taxa in Table 1, shown in Supplementary Figures S1-3.

Viral dereplication resulted in small reference sets, particularly for methods using fixed similarity thresholds. The largest non-hierarchical clustering selection (VSEARCH-0.999) included only 2% of available genomes for lineages with multiple genomes available. In contrast, hierarchical clustering with dynamically determined thresholds yielded more variable reference set sizes, ranging from 1–2% to nearly 60% of all genomes. Because these reference sets are small, overlap between selections is limited (Figure 2**(B)**). Even so, containment indices were consistently higher than expected under random selection, indicating that distinct dereplication strategies do capture partially overlapping subsets of reference genomes.

To further quantify overlap, we assessed whether a "core selection" of reference genomes existed across methods (excluding centroid). We measured this in two ways: (i) the number of genomes shared by all selections, and (ii) the number of species or lineages containing at least one genome selected by every method, restricting analyses to selections that successfully terminated for non-singleton taxa. Despite significant containment indices, no clear viral core selection emerged; only five genomes from distinct lineages were shared by all methods (Supplementary Figure S6). In contrast, bacterial reference sets did have a core selection with approximately 42% of non-singleton species having at least one genome selected by all methods. The size of the bacterial core selection varied between 27 and 169 genomes depending on the experimental setting. When extended to all bacterial species in NCBI, 57% of non-singleton species contributed to the core selection, totaling 2,615 genomes. These results indicate that bacterial genome selections are relatively stable across dereplication methods, whereas viral selections are far more sensitive to the underlying selection strategy.

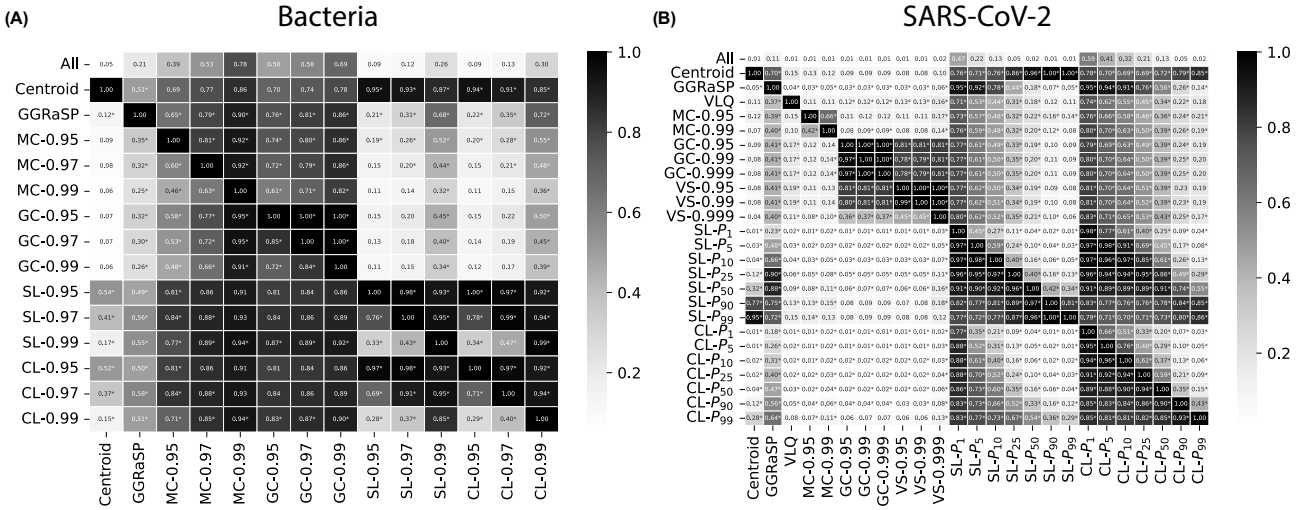


**Figure 2.** Containment indices (CIs) between reference genomes selected by different dereplication tools for the bacterial genomes **(A)** and SARS-CoV-2 genomes (no filtering) **(B)**. CI is only calculated for taxa (species/lineages) that have more than 1 genome to select from. Asterisks indicate that the CI is significantly higher compared to randomly selecting genomes ($p < 0.05$, Benjamini-Hochberg corrected).

### 4.2 Reference genome selection can increase taxonomic profiling accuracy

To investigate the impact of reference genome selection on taxonomic profiling accuracy, we used the constructed reference sets to build taxonomic profiling indexes and analyze simulated metagenomic samples. We evaluate profiling accuracy by calculating abundance accuracy and F1-scores of every sample (see Methods).

*Bacterial experiments*

For each bacterial experiment (genus, family, order), we obtained a sub-database by filtering out species that were not in the selected taxon (Table 1). For example, for the genus experiments we only retained genomes from all the species of the *Streptococcus* genus. This resulted in a sub-database of 1,502 genomes over 103 species for the genus experiments, 12,921 genomes over 381 species for the family experiments, and 14,680 genomes over 691 species for the order experiments. For each experiment, we obtained 15 reference sets (one for each method-threshold combination) and used these for downstream analysis.

Figure 3 shows the abundance accuracy and F1-scores for all bacterial experiments and taxonomic profilers. Across all experiments and profilers, accuracy scores have low variance, indicating consistant outcomes across samples. The "All" reference set consistently ranks among the best performing selections, with median abundance accuracies of approximately 0.9 for Bracken and 0.8 for Centrifuge. Although abundance accuracy across experiments is more variable for DUDes, the "All" selection again ranks among the most accurate (note that DUDes failed to produce any results for the centroid reference set in the genus experiments). While abundance accuracy is relatively stable across experimental settings, F1-scores increase when moving from order-level to genus-level experiments. This trend can be explained by the decreasing number of species in the reference database (14,680, 12,921 and 1502 for order, family and genus respectively), which substantially reduces false positive rates. For both metrics, the "All" reference set consistently ranks among the top-performing configurations, regardless of the experimental setting or taxonomic profiler used.

No single combination of method and similarity threshold consistently outperformed all others. However, selections generated using higher similarity thresholds generally achieved higher accuracy. Among the dereplication approaches, MeShClust and Gclust performed most comparably to the "All" reference set and, in some cases, slightly outperformed it. These findings suggest that while using the full reference database yields highest profiling accuracy, carefully tuned reference genome selection can offer a computationally more efficient alternative with minimal or even no loss in accuracy when sufficiently high similarity thresholds ($\geq 97\%$) are applied.

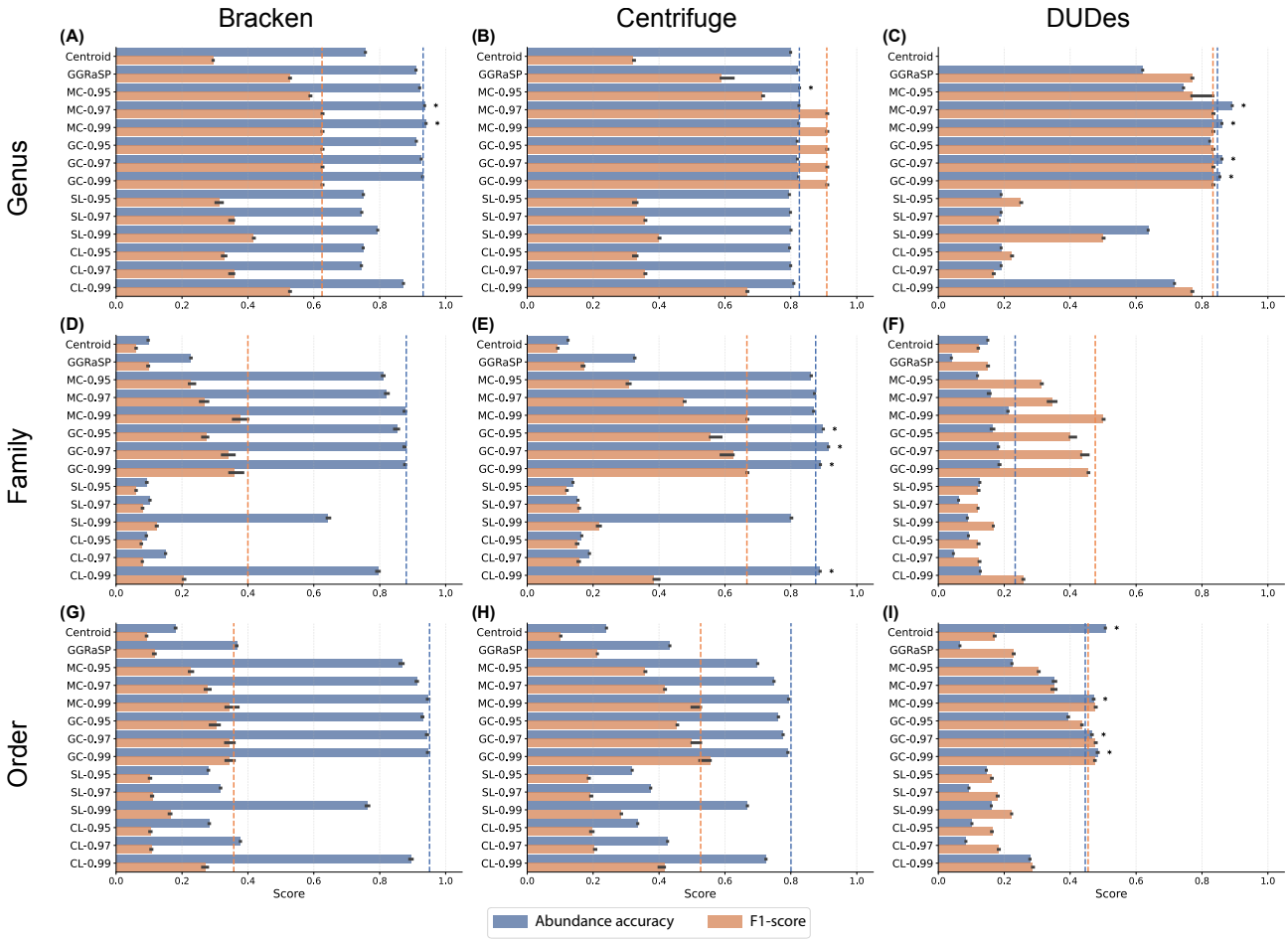


**Figure 3.** Accuracy metrics (abundance accuracy, F1-score) of taxonomic profiling tools in the bacterial experiments. Panels A–I are arranged in three rows (experimental setting: genus, family, order) and three columns (profiling tools: Bracken, Centrifuge, DUDes). Dashed vertical lines correspond to the median performance of the "All" reference set (no selection) and error bars correspond to 10-90th percentile values. An asterisk indicates that the accuracy metric of a method is significantly better than that of the "All" reference set (Wilcoxon signed rank test, $p < 0.05$, Benjamini-Hochberg corrected).

*Viral experiments*

To assess the impact of reference genome selection in a viral context, we focused on SARS-CoV-2 lineage-level abundance estimation. We used a database of 115,850 SARS-CoV-2 genomes spanning 1,020 viral lineages (see Methods), and we obtained 26 reference sets (one for each method-threshold combination) for downstream analysis. Since these genomes are generally more than 99 % similar, this set-up allows us to evaluate dereplication strategies when genetic diversity among reference genomes is extremely low.

Figure 4**(A)** shows that several reference sets achieved significantly higher accuracy than the "All" reference set. Hierarchical clustering-derived reference sets (excluding GGRaSP) consistently performed best; the complete-linkage reference set with the 99th percentile similarity threshold achieved the highest median abundance accuracy (0.72 vs. 0.51 for the "All" set). Other clustering-based selections, including centroid, single-linkage, and comple-linkage sets at various thresholds, also significantly outperformed the "All" reference set in terms of abundance accuracy. By contrast, intermediate thresholds (25th and 50th percentile) for complete-linkage selections yielded no significant improvement. While F1-scores were generally low (a consequence of many false positives due to the high similarity of SARS-CoV-2 genomes), nearly all selections outperform the "All" reference set (median F1-score of 0.095). Overall, these results show that careful reference genome selection can significantly improve viral lineage abundance estimation accuracy.

The results for the viral experiments reveal a different trend than for the bacterial experiments. In the bacterial setting, accuracy was consistent across samples, and including all available genomes generally produced the best results. In contrast, outcome variances were much larger in the viral experiments due to varying sample compositions. Conservative genome selection via hierarchical clustering significantly improved profiling accuracy, even when only a single genome per lineage was retained. This contrast implies that optimal reference set construction is context-dependent, shaped by the diversity and redundancy of the underlying genomes. Together, these findings show that strategies effective for bacterial profiling cannot be directly transferred to viral datasets, and vice versa, without prior evaluation.

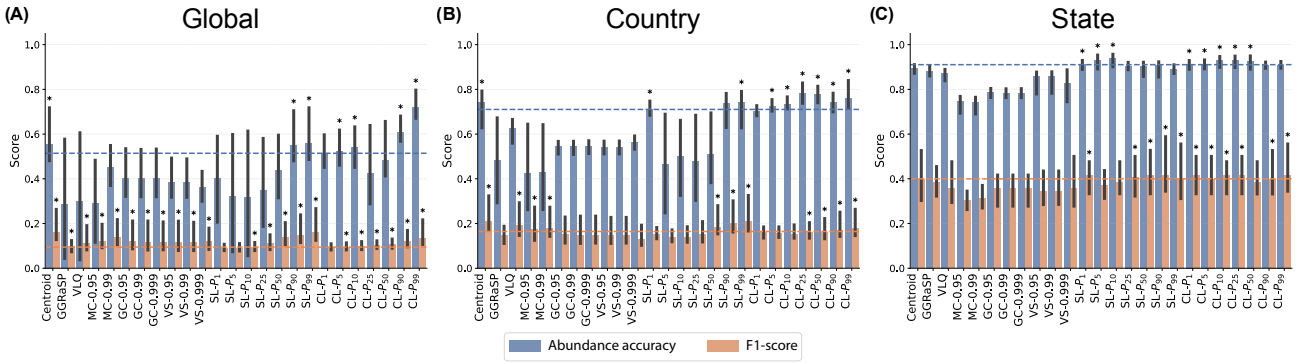


**Figure 4.** Accuracy metrics (abundance accuracy, F1-score) in the SARS-CoV-2 experiments. Panels A–C correspond to the three geographic filtering levels: **(A)** no filter, **(B)** country filter (USA), and **(C)** state filter (Connecticut). Dashed vertical lines correspond to the median performance of the "All" reference set (no selection) and error bars correspond to 10-90th percentile values. An asterisk indicates that the accuracy metric of a method is significantly better than that of the "All" reference set (Wilcoxon signed rank test, $p < 0.05$, Benjamini-Hochberg corrected).

### 4.3 Location-based reference selection improves profiling accuracy for SARS-CoV-2

Given the available metadata for the viral sequences, we investigated whether selecting references based on geographical location can further improve taxonomic profiling accuracy. Our viral samples were simulated to represent SARS-CoV-2 populations in wastewater at a specific point in time in Connecticut, USA. Thus, to evaluate the impact of location-based reference selection, we created reference sets at country and state level, including only sequences from the same country (USA) and state (Connecticut), respectively (see Methods). Following the same workflow as before, we estimated relative abundances of SARS-CoV-2 lineages to assess how geographic filtering affects profiling accuracy.

Figure 4**(B)** and **(C)** show the abundance accuracy and F1-score for all reference sets across all geographic filtering levels. Overall, both abundance accuracy and F1-score benefit from applying increasingly stringent filters: reference sets in the state-filtered experiments produce the most accurate results with the least variance. Notably, the average median abundance accuracy across all methods increased from 0.442 to 0.875 (+109%),

with median F1-scores improved from 0.116 to 0.382 (+240%) on average when comparing state results versus baseline results.

The hierarchical clustering based selections remain consistently more accurate than the "All" selection after applying location filters. In the state-filtered experiments, selections with stricter similarity thresholds performed best, likely reflecting the smaller number of available genomes. As filtering reduced the "All" reference set from 115,850 genomes across 1,020 lineages (global) to 1,678 genomes over 54 lineages (state), a higher similarity threshold was needed to maintain diversity among retained references. Thus, filtering by geographic proximity helps exclude unlikely lineages while focusing on variation among local strains.

These results demonstrate that incorporating context-dependent metadata such as sampling location can complement purely sequence-based reference selection, leading to improved taxonomic profiling accuracy in geographically constrained viral datasets.

### 4.4 Profiling accuracy improves with reference set size for bacteria, but not viruses

In the bacterial experiments we observed that larger reference sets generally result in higher profiling accuracy. In contrast, for our viral experiments this relationship seemed different, as smaller hierarchical clustering-derived reference sets with few genomes performed best. Therefore, we now investigate the impact of reference set size on taxonomic profiling accuracy by correlating the two in both experimental settings.

Figure 5 shows the relationship between the number of included reference genomes and the taxonomic profiling accuracy for all profiling tools in the bacterial experiments. Both measures of accuracy correlate strongly with the number of genomes included in a reference set (Spearman correlation coefficients often exceeding 0.9). However, despite these strong correlations, accuracy gains diminish with the number of included genomes, suggesting that reference sets get saturated as more genomes are included.

For the viral experiments, the strong positive correlations between reference set size and accuracy are largely absent (Figure 6). The F1-score decreases with increasing reference set sizes for both the global and country experiments (Figure 6**(A)**-**(B)**). Additionally, the correlation between reference set size and abundance accuracy is only 0.10 and 0.25 for the global and country experiments, respectively, as smaller reference sets achieve higher abundance accuracy in these experiments.

In the state experiments (Figure 6**(C)**), there is an initial improvement in accuracy when reference sets grow, but this quickly plateaus. When aggregating results across all viral experiments, the Spearman correlations between reference set size and abundance accuracy, and between size and F1-score, were negative (-0.57 and -0.73, respectively). Since the same samples were used in all viral experiments, this shows that smaller viral reference sets tend to achieve higher accuracies. Together, these results support the conclusion that the optimal reference set selection strategy depends on the particular context of experiments.

### 4.5 Computational resource usage analysis

In addition to taxonomic profiling accuracy, another consideration when evaluating reference genome selection methods is their computational resource usage. To assess this, we measured total CPU time (system time + user time) and peak memory usage for all major processing steps (selection, indexing, and profiling). These measurements quantify the trade-off between the upfront cost of genome selection and the potential savings during indexing and repeated profiling. Note that dereplication tools can be run independently for each taxon, hence their effective runtime can be reduced through parallelization.

Figure 7 shows the runtimes for selection and indexing (combined) and taxonomic profiling across all bacterial experiments, grouped by dereplication strategy: hierarchical clustering, MeShClust, Gclust, and All (no selection). Although indexing and profiling times differ between profilers, they show the same trends: hierarchical clustering methods (excluding GGRaSP) result in lower overall runtime and memory usage during selection and indexing compared to the "All" selection, whereas MeShClust and Gclust show higher resource usage (Supplementary Figure S8).

During profiling, runtimes for Bracken and Centrifuge were largely independent of index size, averaging approximately two minutes (Bracken) and 25 minutes (Centrifuge) of processing time per sample. In contrast, DUDes showed a clear relationship between the runtime and index size, with runtimes exceeding nine hours for the "All" reference set in the order experiments, and just under one hour for the centroid reference set in the genus experiments. Peak memory usage scales with index size for all profilers, ranging between 1.5 GB (centroid)

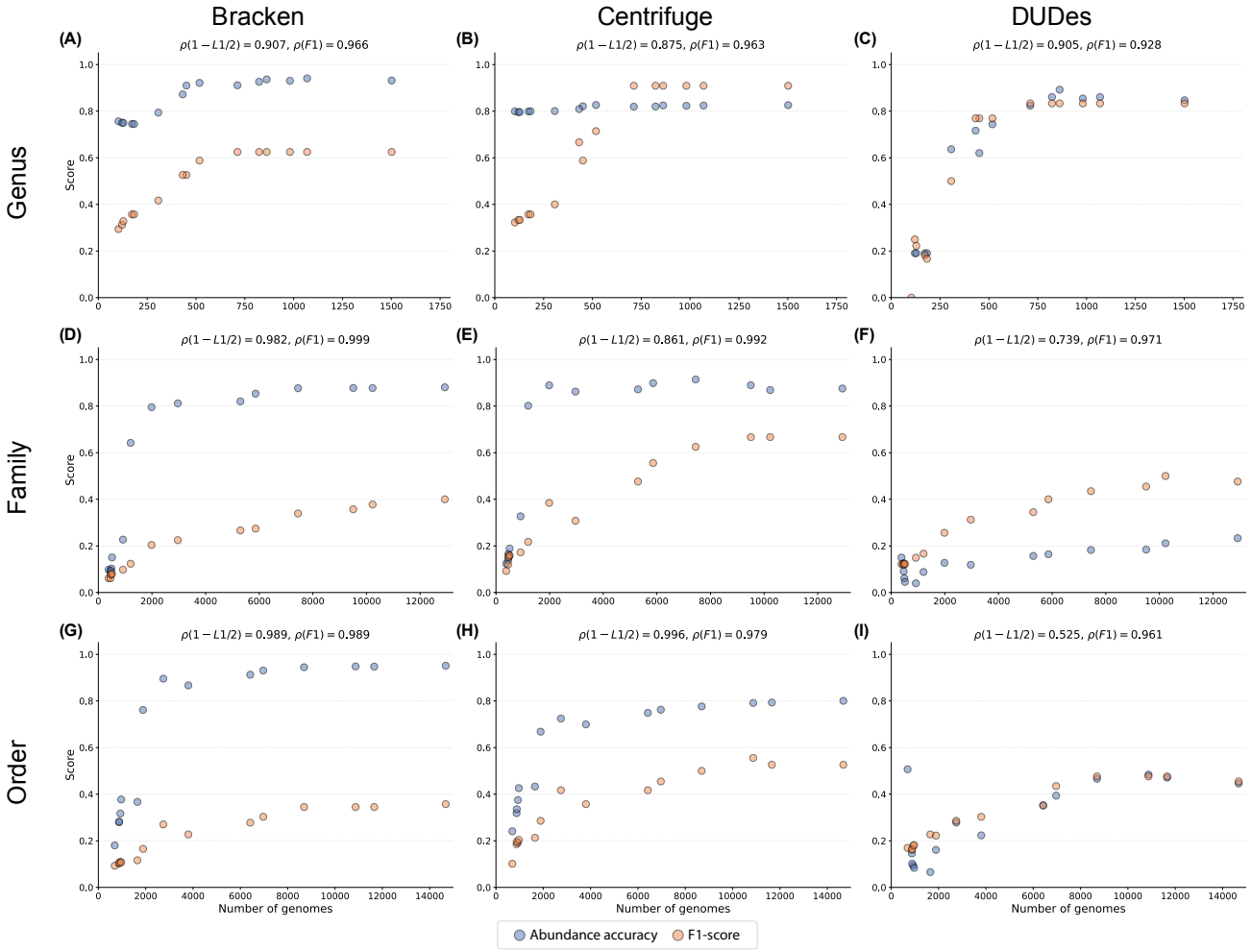


**Figure 5.** Median accuracy (abundance accuracy, F1-score) of taxonomic profiling tools across the bacterial experiments against the number of genomes in a reference set. Panels A–I are arranged in three rows (experimental setting: genus, family, order) and three columns (profiling tools: Bracken, Centrifuge, DUDes). $\rho$-values in subtitles show Spearman correlation between number of genomes included and median accuracy score.

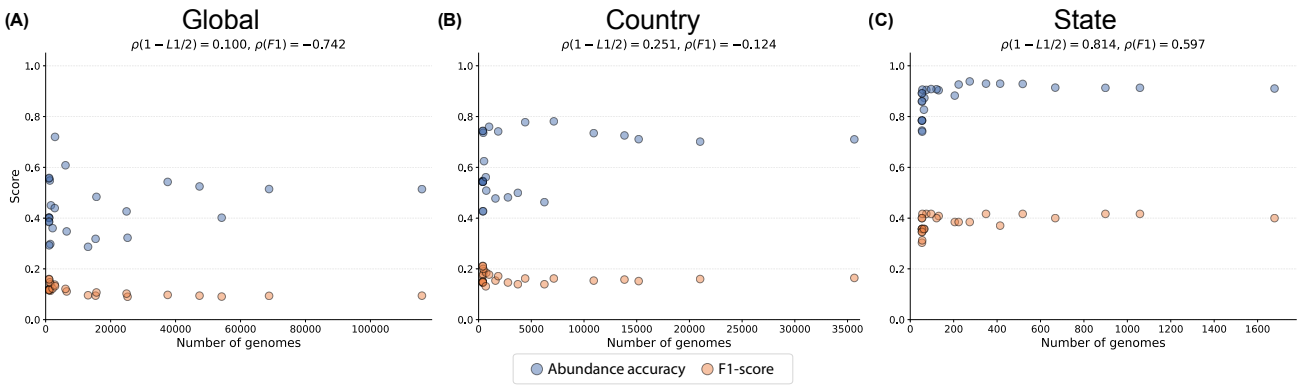


**Figure 6.** Median accuracy (abundance accuracy, F1-score) across SARS-CoV-2 experiments against the number of genomes in a reference set. Panels A–C correspond to the three geographic filtering levels: **(A)** no filter, **(B)** country filter (USA), and **(C)** state filter (Connecticut). $\rho$-values in subtitles show Spearman correlation between number of genomes included and median accuracy score.

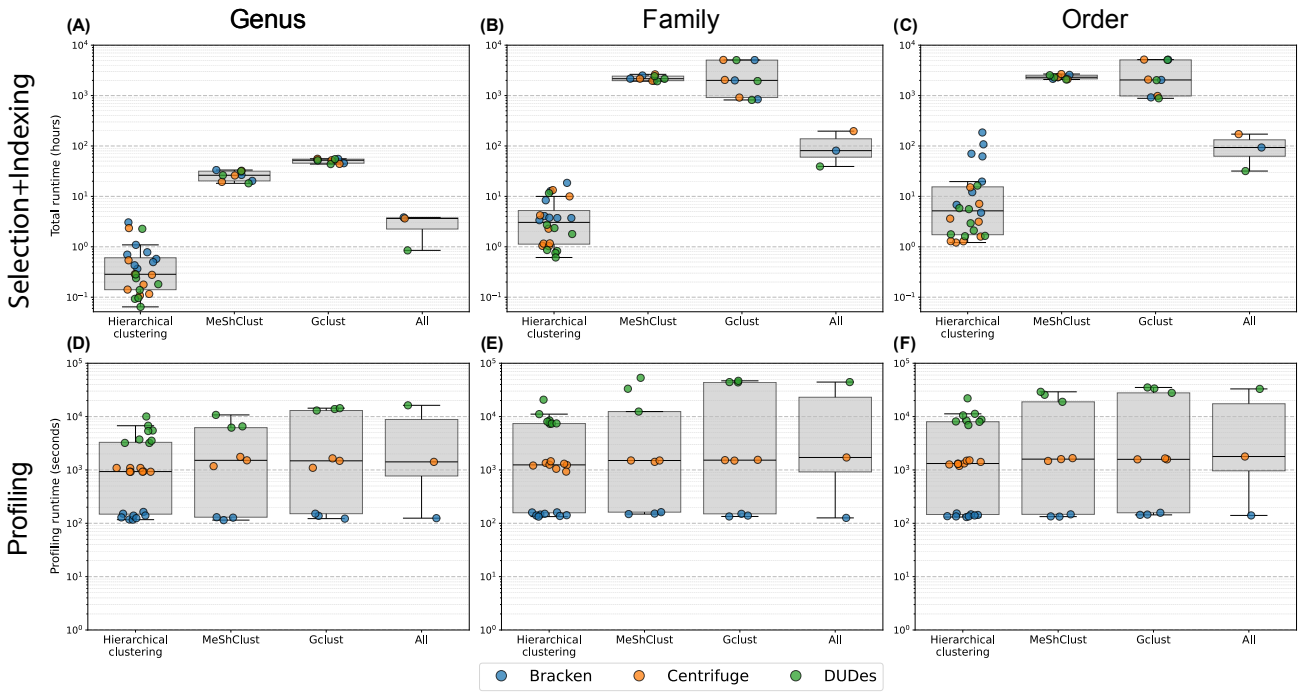


**Figure 7.** Runtime in bacterial experiments for reference selection, indexing, and taxonomic profiling. Dereplication methods are grouped by strategy: hierarchical clustering, MeShClust, Gclust, and All (no selection). Panels A–C show the average combined runtime for selection and indexing, and panels D–F show the average profiling runtime. Columns correspond to the experimental setting (genus, family, order). Profiling tools are color-coded: Bracken (blue), Centrifuge (orange), and DUDes (green). Whiskers in boxplots indicate 25–75th percentile values.

and 32.7 GB ("All") memory for Centrifuge, and between 8.2 GB (centroid) and 163.8 GB ("All") memory for DUDes in the order experiments. For Bracking, this effect was modest, with memory usage ranging between 3.1 GB (centroid) and 4.6 GB ("All"). Thus, while reference selection incurs an initial computational cost, there is a clear benefit in terms of peak memory usage during profiling.

In the viral experiments (Figure 8), indexing strongly scaled with reference set size, with the "All" reference set requiring the most time and memory (Supplementary Figure S9). However, when including the selection stage, other reference sets took longer overall. Peak memory usage scaled with reference set size, except for the VLQ-based selection which had a roughly constant memory footprint across experiments, resulting in relatively higher memory usage in the smaller state-filtered datasets (Supplementary Tables S4-6). Profiling runtimes were again determined primarily by index size, with the "All" selection consistently being the most resource-intensive.

Overall, reference selection primarily benefits memory efficiency in bacterial profiling, while in the viral setting, both indexing and profiling times improved substantially due to the smaller number of reference genomes. This highlights that the computational benefits of reference genome selection depend on dataset size, genomic redundancy, and profiling method.

## 5 Discussion

As genome databases continue to grow, identifying representative reference genomes has become essential to ensure computationally efficient downstream analyses [22]. Here, we systematically evaluated sequence dereplication methods for selecting representative viral and bacterial reference genomes and quantified their impact on taxonomic profiling accuracy and computational performance. We estimated species abundances for bacterial datasets, and lineage abundances for viral datasets, finding that reference genome selection can substantially improve profiling accuracy and computational efficiency in the viral experiments. In contrast, using all available bacterial reference genomes (i.e. without selection) yielded slightly higher accuracy without a major increase in computational cost compared to using dereplication strategies.

We believe that the gap between the effects observed in the bacterial and viral experiments is driven by fundamental differences in genomic diversity and redundancy, and taxonomic resolution. In the viral experiments, abundance estimation was performed at the lineage level, which is intrinsically harder than the species-level

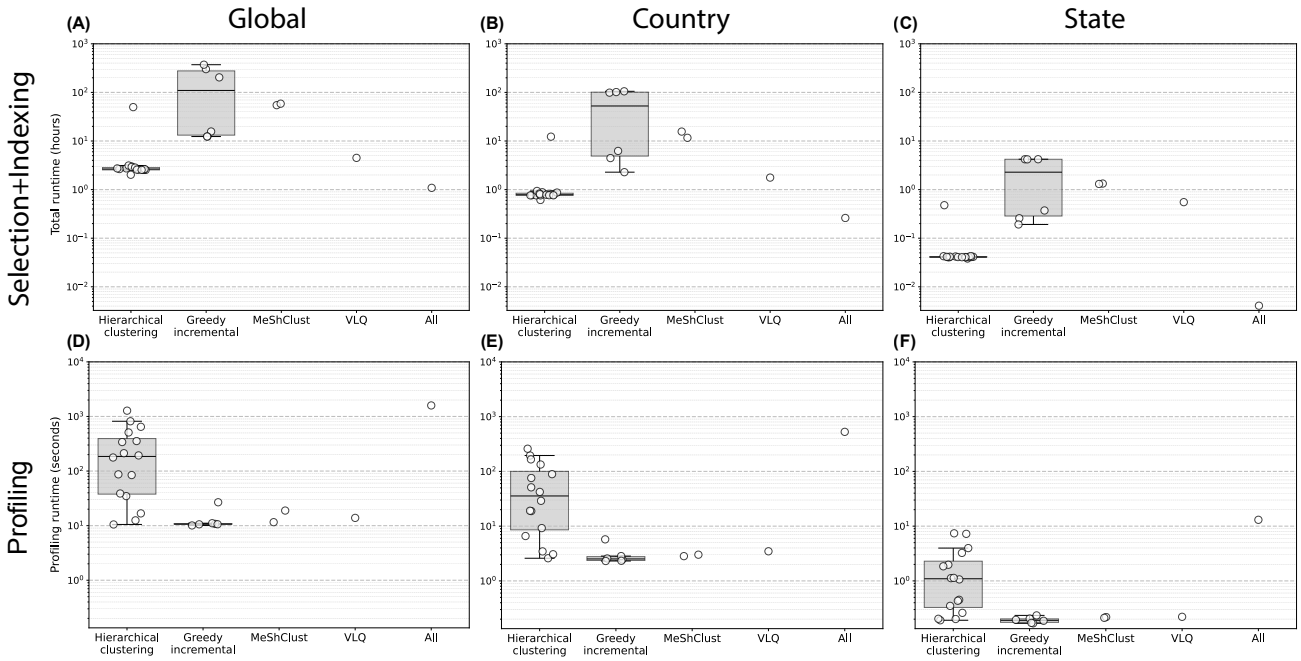


**Figure 8.** Runtime in SARS-CoV-2 experiments for reference selection, indexing, and taxonomic profiling. Dereplication methods are grouped by strategy: hierarchical clustering, MeShClust, Gclust, VLQ, and All (no selection). Panels A–C show the average combined runtime for selection and indexing, and panels D–F show the average profiling runtime. Columns correspond to the experimental setting (no filter, country filter, state filter). Whiskers in boxplots indicate 25–75th percentile values.

estimation for the bacterial experiments due to higher similarities. Moreover, sequence similarity among SARS-CoV-2 genomes of the same lineage is much higher than that of bacterial genomes of the same species. As a consequence, distinguishing between SARS-CoV-2 lineages is inherently more difficult, which may explain the greater accuracy gains obtained from reference genome selection in the viral experiments.

Beyond sequence similarity-based dereplication, our viral experiments also emphasize the utility of incorporating contextual metadata in reference genome selection. Previous work has shown that including measures related to completeness can aid in selecting representative sequences [35]. Our results extend this notion, showing that incorporating geographical proximity to samples further improves profiling accuracy, as lineages are often associated with specific regions.

Reference genome selection introduces a clear computational trade-off whose benefits depend on context. In the bacterial experiments, the selection step added computational cost without improving profiling accuracy or substantially reducing resource usage. Moreover, the dereplication strategies that achieved the highest accuracy were also the slowest, while faster strategies performed worse. This aligns with the observation that larger reference sets yielded better bacterial profiling results [22]. In contrast, in the viral experiments, selection dramatically reduced profiling runtime and memory usage by up to two orders of magnitude. Here, the smallest reference sets not only led to the fastest profiling but also produced the highest accuracy, reflecting the redundancy among closely related SARS-CoV-2 genomes and the advantages of targeted selection. Together, these results show that both biological and computational factors shape when reference genome selection is beneficial, and that optimal reference set design depends on the resolution, diversity, and scale of the analysis.

An inherent difficulty of performing experiments at this scale is the computational resource requirements when constructing large taxonomic profiling indexes [21]. Consequently, we were limited to smaller-scale bacterial experiments as constructing profiling indices using all available reference genomes, as well as large bacterial reference sets such as Gclust with a 99 % similarity threshold would have required terabytes of memory. Ideally, large scale experiments, such as those in the CAMI challenges [18, 19] should be performed to more comprehensively assess the impact of reference genome selection on taxonomic profiling. Such experiments would also make it possible to further test the hypothesis that the utility of selection indeed depends on taxon similarity, as we have seen in our experiments.

While several dereplication methods were shown to have a positive impact on taxonomic profiling accuracy, these methods depend strongly on a similarity threshold. Using a threshold of 99% struck a fine balance between the number of genomes selected and the taxonomic profiling accuracy achieved by Gclust and MeShClust, while

the same threshold worked poorly for MASH-based approaches in our bacterial experiments. Conversely, fixed thresholds for Gclust, MeShClust, and VSEARCH yielded poor results in the viral experiments, where the best performance was achieved with hierarchical clustering-based selections using data-driven, adaptive similarity thresholds. This suggests that threshold optimization should be treated as an integral part of reference genome selection rather than an arbitrary choice.

Given these limitations, a key next step is the development of reference selection methods designed specifically for taxonomic profiling. Although existing dereplication tools can be applied per taxon to ensure at least one genome is retained, this approach ignores inter-taxon similarities—an important factor for accurate classification. Developing methods that explicitly optimize discrimination between taxa could therefore improve both the accuracy and efficiency of taxonomic profiling, ultimately enabling more scalable and reliable analyses of microbial communities.

## 6 Additonal Files

**Supplementary Figure S1** Containment indices (CI) between reference genomes for the bacterial genus experiments. CI is only calculated for species that have more than one genome to select from. Asterisks indicate that the CI is significantly higher compared to randomly selecting genomes ($p \leq 0.05$, Benjamini-Hochberg corrected).

**Supplementary Figure S2** Containment indices (CI) between reference genomes for the bacterial family experiments. CI is only calculated for species that have more than one genome to select from. Asterisks indicate that the CI is significantly higher compared to randomly selecting genomes ($p \leq 0.05$, Benjamini-Hochberg corrected).

**Supplementary Figure S3** Containment indices (CI) between reference genomes for the bacterial order experiments. CI is only calculated for species that have more than one genome to select from. Asterisks indicate that the CI is significantly higher compared to randomly selecting genomes ($p \leq 0.05$, Benjamini-Hochberg corrected).

**Supplementary Figure S4** Containment indices (CI) between reference genomes for the viral country experiments. CI is only calculated for lineages that have more than one genome to select from. Asterisks indicate that the CI is significantly higher compared to randomly selecting genomes ($p \leq 0.05$, Benjamini-Hochberg corrected).

**Supplementary Figure S5** Containment indices (CI) between reference genomes for the viral state experiments. CI is only calculated for lineages that have more than one genome to select from. Asterisks indicate that the CI is significantly higher compared to randomly selecting genomes ($p \leq 0.05$, Benjamini-Hochberg corrected).

**Supplementary Figure S6** Core selection analysis for viral experiments. Top row: number of lineages that have at least one genome selected by every method in the viral experiments. Bottom row: number of genomes selected by every method in the viral experiments.

**Supplementary Figure S7** Core selection analysis for bacterial experiments. Top row: number of species that have at least one genome selected by every method in the bacterial experiments. Bottom row: number of genomes selected by every method in the bacterial experiments.

**Supplementary Figure S8** Memory usage for bacterial experiments. **(A), (B)** and **(C)** show peak memory usage during selection and indexing for genus, family and order experiments respectively. **(D), (E)** and **(F)** show peak memory usage during profiling for genus, family and order experiments respectively.

**Supplementary Figure S9** Memory usage for viral experiments. **(A), (B)** and **(C)** show peak memory usage during selection and indexing for global, country and state experiments respectively. **(D), (E)** and **(F)** show peak memory usage during profiling for global, country and state experiments respectively.

**Supplementary Table S1** Computational resource usage for bacterial genus experiments (including alignment statistics).

**Supplementary Table S2** Computational resource usage for bacterial family experiments (including alignment statistics).

**Supplementary Table S3** Computational resource usage for bacterial order experiments (including alignment statistics).

**Supplementary Table S4** Computational resource usage for viral global experiments (including alignment statistics).

**Supplementary Table S5** Computational resource usage for viral country experiments (including alignment statistics).

**Supplementary Table S6** Computational resource usage for viral state experiments (including alignment statistics).

## 7 Availability of source code and requirements

Supporting scripts as well as instructions for obtaining the results presented in this manuscript are available on Github[55].

- Project name: reference_set_selection_benchmark
- Project homepage: `https://github.com/JaspervB-tud/reference_set_selection_benchmark`
- Archived version: 10.5281/zenodo.17544959
- Operating system(s): Linux
- Programming language: Python, Shell and R
- License: MIT

## 8 Data availability

The datasets supporting the results of this article are available in a Zenodo repository [56] (10.5281/zenodo.17544966).

## 9 Declarations

### 9.1 List of abbreviations

BP: Basepairs; CI: Containment Index; GTDB: Genome Taxonomy Database; Mbp: Megabase pair; NCBI: National Center for Biotechnology Information; VLQ: Viral Lineage Quantification

### 9.2 Consent for publication

Not applicable.

### 9.3 Competing Interests

The authors declare that they have no competing interests.

### 9.4 Funding

This research received no external funding.

### 9.5 Author's Contributions

Jasper van Bemmelen (Conceptualization; Methodology; Software; Validation; Formal analysis; Investigation; Data curation; Visualization; Writing–original draft; Writing–review & editing), Ioanna Nika (Methodology; Software; Validation; Formal analysis; Investigation; Data curation; Writing–original draft; Writing–review), Jasmijn A. Baaijens (Conceptualization; Supervision; Project administration; Writing–review & editing).

## 10 Acknowledgements

We gratefully acknowledge all data contributors, i.e., the Authors and their Originating laboratories responsible for obtaining the specimens, and their submitting laboratories for generating the genetic sequence and metadata and sharing via the GISAID Initiative, on which part of this research is based.

Research reported in this work was partially or completely facilitated by computational resources and support of the Delft AI Cluster (DAIC) at TU Delft (RRID: SCR_025091), but remains the sole responsibility of the authors, not the DAIC team.

## References

[1] David L Wheeler, Tanya Barrett, et al. Database resources of the national center for biotechnology information. *Nucleic acids research*, 36(suppl_1):D13–D21, 2007.

[2] Donovan H Parks, Maria Chuvochina, et al. GTDB: an ongoing census of bacterial and archaeal diversity through a phylogenetically consistent, rank normalized and complete genome-based taxonomy. *Nucleic acids research*, 50(D1):D785–D794, 2022.

[3] Derrick E Wood, Jennifer Lu, and Ben Langmead. Improved metagenomic analysis with kraken 2. *Genome biology*, 20:1–13, 2019.

[4] Rachid Ounit, Steve Wanamaker, et al. CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC genomics*, 16:1–13, 2015.

[5] Daehwan Kim, Li Song, et al. Centrifuge: rapid and sensitive classification of metagenomic sequences. *Genome research*, 26(12):1721–1729, 2016.

[6] Aitor Blanco-Míguez, Francesco Beghini, et al. Extending and improving metagenomic taxonomic profiling with uncharacterized species using MetaPhlAn 4. *Nature Biotechnology*, 41(11):1633–1644, 2023.

[7] Hans-Joachim Ruscheweyh, Alessio Milanese, et al. mOTUs: profiling taxonomic composition, transcriptional activity and strain populations of microbial communities. *Current Protocols*, 1(8):e218, 2021.

[8] Kim D Pruitt, Tatiana Tatusova, and Donna R Maglott. NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic acids research*, 35(suppl_1): D61–D65, 2007.

[9] Tamara Goldfarb, Vamsi K Kodali, et al. NCBI RefSeq: reference sequence standards through 25 years of curation and annotation. *Nucleic Acids Research*, 53(D1):D243–D257, 2025.

[10] David Koslicki, Stephen White, et al. YACHT: an ANI-based statistical test to detect microbial presence/absence in a metagenomic sample. *Bioinformatics*, 40(2):btae047, 2024.

[11] Lucas R van Dijk, Bruce J Walker, et al. StrainGE: a toolkit to track and characterize low-abundance strains in complex microbial communities. *Genome biology*, 23(1):74, 2022.

[12] Limin Fu, Beifang Niu, et al. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 28(23):3150–3152, 2012.

[13] Robert C Edgar. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, 26(19): 2460–2461, 2010.

[14] Hani Z Girgis. MeShClust v3. 0: high-quality clustering of DNA sequences using the mean shift algorithm and alignment-free identity scores. *BMC genomics*, 23(1):423, 2022.

[15] Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

[16] Josip Marić, Krešimir Križanović, et al. Comparative analysis of metagenomic classifiers for long-read sequencing datasets. *BMC bioinformatics*, 25(1):15, 2024.

[17] Alexa BR McIntyre, Rachid Ounit, et al. Comprehensive benchmarking and ensemble approaches for metagenomic classifiers. *Genome biology*, 18:1–19, 2017.

[18] Alexander Sczyrba, Peter Hofmann, et al. Critical assessment of metagenome interpretation—a benchmark of metagenomics software. *Nature methods*, 14(11):1063–1071, 2017.

[19] Fernando Meyer, Adrian Fritz, et al. Critical assessment of metagenome interpretation: the second round of challenges. *Nature methods*, 19(4):429–440, 2022.

[20] H Ye Simon, Katherine J Siddle, et al. Benchmarking metagenomics tools for taxonomic classification. *Cell*, 178(4):779–794, 2019.

[21] Daniel J Nasko, Sergey Koren, et al. RefSeq database growth influences the accuracy of k-mer-based lowest common ancestor species identification. *Genome biology*, 19:1–10, 2018.

[22] Robyn J Wright, Andrè M Comeau, and Morgan GI Langille. From defaults to databases: parameter and database choice dramatically impact the performance of metagenomic taxonomic classification tools. *Microbial Genomics*, 9(3):000949, 2023.

[23] Ruijie Xu, Sreekumari Rajeev, and Liliana CM Salvador. The selection of software and database for metagenomics sequence analysis impacts the outcome of microbial profiling and pathogen detection. *Plos one*, 18 (4):e0284031, 2023.

[24] Vanessa R. Marcelino, Edward C Holmes, et al. The use of taxon-specific reference databases compromises metagenomic classification. *BMC genomics*, 21(1):184, 2020.

[25] Quan Zou, Gang Lin, et al. Sequence clustering in bioinformatics: an empirical study. *Briefings in bioinformatics*, 21(1):1–10, 2020.

[26] Ze-Gang Wei, Xu Chen, et al. Comparison of methods for biological sequence clustering. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(5):2874–2888, 2023.

[27] Zhen Ju, Huiling Zhang, Jingjing Zhang, Wenhui Xi, Dian Huang, Shengzhong Feng, Jintao Meng, and Yanjie Wei. A comparative study of sequence clustering algorithms. *Big Data Mining and Analytics*, 8(5): 1011–1022, 2025.

[28] Torbjørn Rognes, Tomáš Flouri, et al. Vsearch: a versatile open source tool for metagenomics. *PeerJ*, 4: e2584, 2016.

[29] Ruilin Li, Xiaoyu He, et al. Gclust: A parallel clustering tool for microbial genomic data. *Genomics, Proteomics and Bioinformatics*, 17(5):496–502, 2019.

[30] Thomas H Clarke, Lauren M Brinkac, et al. GGRaSP: a r-package for selecting representative genomes using gaussian mixture models. *Bioinformatics*, 34(17):3032–3034, 2018.

[31] Alexandr Andoni and Robert Krauthgamer. The computational hardness of estimating edit distance. *SIAM Journal on Computing*, 39(6):2398–2429, 2010.

[32] Maxwell W Libbrecht, Jeffrey A Bilmes, and William Stafford Noble. Choosing non-redundant representative subsets of protein sequence data sets using submodular optimization. *Proteins: Structure, Function, and Bioinformatics*, 86(4):454–466, 2018.

[33] Konstantinos G Derpanis. Mean shift clustering. *Lecture Notes*, 32(1-4):16, 2005.

[34] Benjamin T James, Brian B Luczak, and Hani Z Girgis. MeShClust: an intelligent tool for clustering dna sequences. *Nucleic acids research*, 46(14):e83–e83, 2018.

[35] Matthew R Olm, Christopher T Brown, et al. dRep: a tool for fast and accurate genomic comparisons that enables improved genome recovery from metagenomes through de-replication. *The ISME journal*, 11(12): 2864–2868, 2017.

[36] Brian D Ondov, Todd J Treangen, et al. Mash: fast genome and metagenome distance estimation using MinHash. *Genome biology*, 17:1–14, 2016.

[37] Chirag Jain, Luis M Rodriguez-R, Adam M Phillippy, Konstantinos T Konstantinidis, and Srinivas Aluru. High throughput ani analysis of 90k prokaryotic genomes reveals clear species boundaries. *Nature communications*, 9(1):5114, 2018.

[38] Martin Steinegger and Johannes Söding. Clustering huge protein sequence sets in linear time. *Nature communications*, 9(1):2542, 2018.

[39] Alexey Markin, Sanket Wagle, Siddhant Grover, Amy L Vincent Baker, Oliver Eulenstein, and Tavis K Anderson. Parnas: objectively selecting the most representative taxa on a phylogeny. *Systematic Biology*, 72(5):1052–1063, 2023.

[40] Simone Marini, Carla Mavian, Alberto Riva, Mattia Prosperi, Marco Salemi, and Brittany Rife Magalis. Optimizing viral genome subsampling by genetic diversity and temporal distribution (tardis) for phylogenetics. *Bioinformatics*, 38(3):856–860, 2022.

[41] Wei Zhou, Nicole Gay, and Julia Oh. Reprdb and pandb: minimalist databases with maximal microbial representation. *Microbiome*, 6:1–15, 2018.

[42] Jasmijn A Baaijens, Alessandro Zulli, et al. Lineage abundance estimation for SARS-CoV-2 in wastewater using transcriptome quantification techniques. *Genome biology*, 23(1):236, 2022.

[43] Jennifer Lu, Florian P Breitwieser, et al. Bracken: estimating species abundance in metagenomics data. *PeerJ Computer Science*, 3:e104, 2017.

[44] Vitor C Piro, Martin S Lindner, and Bernhard Y Renard. DUDes: a top-down taxonomic profiler for metagenomics. *Bioinformatics*, 32(15):2272–2280, 2016.

[45] Konstantinos T Konstantinidis and James M Tiedje. Genomic insights that advance the species definition for prokaryotes. *Proceedings of the National Academy of Sciences*, 102(7):2567–2572, 2005.

[46] Jasper van Bemmelen. reference_set_selection_benchmark. `https://github.com/JaspervB-tud/reference_set_selection_benchmark`, 2025. GitHub repository.

[47] Fabian Pedregosa, Gaël Varoquaux, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[48] Heng Li and Richard Durbin. Fast and accurate short read alignment with Burrows–Wheeler transform. *bioinformatics*, 25(14):1754–1760, 2009.

[49] Zheng Sun, Shi Huang, et al. Challenges in benchmarking metagenomic profilers. *Nature methods*, 18(6): 618–626, 2021.

[50] Weichun Huang, Leping Li, et al. ART: a next-generation sequencing read simulator. *Bioinformatics*, 28 (4):593–594, 2012.

[51] Shruti Khare, Céline Gurry, et al. GISAID's role in pandemic response. *China CDC weekly*, 3(49):1049, 2021.

[52] Nicolas L Bray, Harold Pimentel, et al. Near-optimal probabilistic RNA-seq quantification. *Nature biotechnology*, 34(5):525–527, 2016.

[53] Delft AI Cluster (DAIC). The delft ai cluster (daic), rrid:scr_025091, 2024. URL `https://doc.daic.tudelft.nl/`.

[54] Fernando Meyer, Andreas Bremges, et al. Assessing taxonomic metagenome profilers with OPAL. *Genome biology*, 20:1–10, 2019.

[55] Jasper van Bemmelen. Reference selection repository. https://github.com/JaspervBtud/reference_set_selection_benchmark, 2025.

[56] Jasper van Bemmelen, Jasmijn A. Baaijens, and Ioanna Nika. Simulated reads and genome accession ids, January 2025.