# 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*
[*] *Corresponding authors: j.vanbemmelen-1@tudelft.nl; j.a.baaijens@tudelft.nl*

**Abstract**

**Motivation:** 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 filtering the input genome database, there is limited work exploring the impact of different sequence dereplication methods across taxonomic classification tools.
**Results:** We assess the effect of genome selection approaches on taxonomic classification using both bacterial and viral datasets. Our results demonstrate that careful selection of reference genomes can improve classification accuracy. We also show that using prior knowledge about a metagenomic sample, such as sampling location, can significantly improve classification accuracy. Finally, we find that using a redundancy-filtered genome database generally reduces the computational resources required, with minimal loss in classification accuracy.

**Keywords:** Taxonomic classification, Metagenomics, Sequence clustering, Genome selection, Sequence dereplication

## 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) [42] and the Genome Taxonomy Database (GTDB) [27] doubling in size every few years. As taxonomic profiling inherently relies on the use of taxonomic databases, this means that taxonomic profiling tools have to cope with the explosive growth of genome databases.

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 [43], 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 needed to run the tool, compared to its predecessor. In a similar fashion, CLARK [26] only stores $k$-mers that are unique to the genomes in the database. Instead of using a compressed $k$-mer-based representation, Centrifuge [15] progressively builds a pangenome for every species by adding non-redundant segments of genomes. These pangenomes are stored in an FM-index-based [9] database. Marker-based taxonomic profiling tools, like MetaPhlan [3] and mOTUs [33], use databases which only store marker gene sequences.

Rather than solely relying on database compression techniques, taxonomic profiling can also benefit from carefully selecting representative genomes. This approach not only improves scalability by reducing database sizes, but also has the potential to improve profiling accuracy by minimizing false positives. Despite these advantages, our understanding of optimal strategies for genome selection is limited. Recent methods [16, 40] incorporate a selection step to filter out highly similar genomes during index generation, which reduces false positive rates [16]. However, it remains unclear how widely applicable such selection strategies are, or whether tools like CD-HIT [10], mmseqs2 [36], or MeShClust [11]—designed for sequence dereplication—can further improve taxonomic profiling pipelines through reference genome selection.

Genome databases have also focused on establishing non-redundant reference sets by selecting curated, high-quality genome sequences. A primary example is NCBI's RefSeq [30], one of the most widely used genome databases, which includes representative genomes from over 28,000 named species over all taxonomic kingdoms [12]. This database serves as a standard for many taxonomic profiling tools, including only a single

(representative) genome per taxonomic level. 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 also define species boundaries within the taxonomy and are updated as higher-quality genomes become available [27]. These curated databases can play an important role in enabling accurate and efficient taxonomic profiling.

To maximize taxonomic profiling accuracy, it is essential to understand how reference genome selection impacts results. While several benchmarking studies have compared the performance of different taxonomic profiling tools ([20, 21, 34, 22, 35]), 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 [23] or other standard databases in combination with different profiling parameters [44]. Other work has focused on the use of taxon-specific reference sets, highlighting how this can bias results [31]. 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 in understanding, 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 selection methods, including VSEARCH [32], MeShClust [11], Gclust [18], and GGRaSP [5], as well as hierarchical clustering approaches. 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 genomes are highly similar, filtered reference sets can significantly improve profiling accuracy while substantially reducing computational resource requirements. This highlights the value of reference genome selection in optimizing taxonomic profiling.

# 2 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 existing dereplication tools, followed by a description of datasets, benchmarking setup, and performance evaluation.

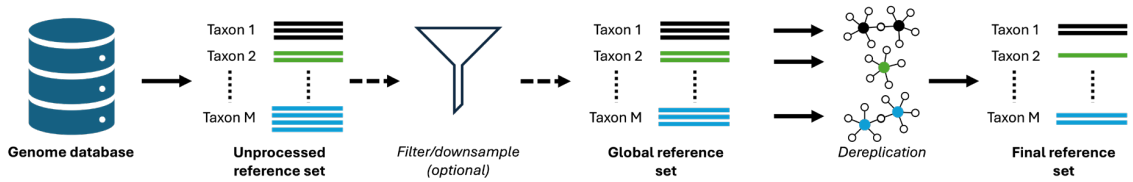


Figure 1: A schematic representation of the reference genome selection pipeline. Potential reference genomes are retrieved from a database. Optional quality filtering and/or downsampling can be applied. Sequence dereplication methods are applied to the global reference set with genomes for every taxon of interest, resulting in a final reference set.

## 2.1 Overview of sequence dereplication methods

The problem that these tools solve is to, given a set of sequences $S$ and similarity function $f$ and a similarity threshold $T$, find a (small) subset $\overline{S} \subseteq S$ such that for every sequence $s \in S$ there exists at least one sequence $\overline{s} \in \overline{S}$ with $f(s, \overline{s}) \geq T$. The key idea here, is that every sequence in $S$ is represented by at least one sequence in $\overline{S}$, thereby reducing redundancy.

As sequence dereplication tools have primarily been developed to operate on reads, they are designed to deal with large amounts of sequences. To achieve this, these tools generally attempt to minimize the number of sequence comparisons that have to be made, since comparing sequences based on alignment is computationally expensive [1], and the number of comparisons scales quadratically with the number of sequences. This has led to a general framework, often called *greedy incremental clustering* in which sequences are sorted according to a total order and selecting the first sequence as a representative of its own sequence cluster. Afterwards, subsequent sequences are compared to representatives, adding them to an existing cluster if their similarity with the corresponding representative is larger than the given threshold $T$, or becoming the representative of their own cluster if they are dissimilar to all other representatives.

Many popular sequence dereplication tools, like CD-HIT [10], UCLUST [8], VSEARCH [32], mmseqs2 [36], LINCLUST [37] and Gclust [18] utilize this framework. Despite using the same general framework, however, these methods have specific implementations and differ, for example, 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.

In addition to tools that implement the greedy incremental clustering framework, other methods implement different clustering techniques to find representative sequences. For example, MeShClust [11] proposes using a mean-shift [7] algorithm for clustering, together with a generalized linear model that is trained to classify pairs of sequences as being similar or not (according to a fixed threshold $T$). dRep [24] and GGRaSP [5] instead use a strategy similar to that used by StrainGE [40] and YACHT [16], in which they perform hierarchical clustering, and select cluster medoids as representatives.

While there are some papers that have benchmarked several of these methods (e.g. [45, 41]), these benchmarks are often limited to using these tools on large sets of relatively short sequences. Out of the methods listed here, only MeShClust, dRep, Gclust and GGRaSP have been shown to work on full length bacterial genomes. Since our goal is to perform sequence selection on viral and bacterial genomes, we will therefore use these four tools. Additionally, we include VSEARCH for part of our experiments to highlight the computational complexity of exact alignment based approaches.

## 2.2 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, while for the SARS-CoV-2 scenario, the goal is to classify sequence at the lineage level. This setup enables us to investigate whether the impact of reference selection depends on the similarity of the target genomes. Given the differences in resolution required for these scenarios, we used different profiling methods: generic taxonomic classification tools (Kraken2 [43] + Bracken [19], Centrifuge [15], BWA [17] + DUDes [29]) for the bacterial scenario, and the VLQ pipeline [2] for the SARS-CoV-2 scenario. We describe the methodology for each scenario in more detail below.

### 2.2.1 Bacteria experiments

For the bacterial scenario, we perform three types of experiments based on NCBI's taxonomy (downloaded September 24th, 2024): i) bacterial species of the same order, ii) bacterial species of the same family, and iii) bacterial species of the same genus. We will refer to these settings as "order experiments", "family experiments", and "genus experiments", respectively. In each experiment, we perform 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 dRep.

All tools were run with similarity thresholds of 95%, 97%, and 99%, except GGRaSP which uses an automated threshold selection strategy. The genomes that were provided to these tools consisted of all "Complete Genome" status assemblies from NCBI Assembly (downloaded on October 9th, 2024), considering only species that had at least one such assembly. For tools that required genomes to be provided in a single multi-fasta file, all genomic segments were concatenated, and the resulting genomes were added to a single multi-fasta file.

The order, family and genus are shown in Table 1 and were selected as follows: first we remove any species that have fewer than 100 genomes in the database in order to be able to select reference genomes. Afterwards, we sort the remaining species based on the average intra-species distance (estimated using MASH as explained below), from highest to lowest. For all experiments, the corresponding taxon that was chosen, was the first for which we were able to find 5 species by parsing the sorted species list. Although we select 5 species from every selected taxon, all other species of the selected taxon with at least one "Complete Genome" assembly were also processed when running the dereplication tools.

The parameters used to run every tool can be found on Github, with the number of threads used shown in Supplementary Table S1. For both GGRaSP and our implementation of dRep we used MASH (v2.3) [25] 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 our implementation of dRep, we only performed a single round of hierarchical clustering (using scikit-learn v1.5.0 [28]) based on the computed MASH distances, generating both a reference set using single-linkage clustering and complete-linkage clustering at the designated thresholds, and selecting the centroid of each cluster as its representative. In addition we also generate a "centroid" reference set for every species, containing only the centroid genome for every species, and an unprocessed "All" reference set that contains all genomes for every species.

For each reference set, we build indexes for Kraken2 (v2.1.3) in combination with Bracken (v1.0.0), Centrifuge (v1.0.4.2) and BWA (v0.7.18) in combination with DUDes (v0.10.0). With this selection of methods, we cover 2 popular $k$-mer based taxonomic profiling tools with different methods for compressing reference genomes (Kraken2+Bracken, Centrifuge) as well as one alignment based tool (BWA+DUDes). For all methods, the reference indexes were built using the default parameters provided by the methods.

A complicating factor is that these taxonomic profiling tools do not estimate taxon abundances in the same way. Specifically, Kraken2 and Bracken estimate "sequence abundance", 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 notions of abundance, is that taxonomic abundances incorporate the lengths of reference genomes, and thus these outcomes cannot be compared directly [38]. Centrifuge estimates both a sequence abundance, as well as a taxonomic abundance.

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) [13]. For all bacterial scenarios, 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 paired-end reads generated from 10 genomic assemblies from NCBI Assembly that have a "Scaffold" or "Chromosome" quality, selecting the top 10 assemblies per species based on completeness. Throughout all experiments, the sequence abundance of every species is fixed at approximately 20% (uniformly distributed over the 10 genomes of every species), but taxonomic abundances vary based on genome lengths.

Table 1: List of the selected genus, family and order taxons, as well as the corresponding species selected for every experiment.

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

#### 2.2.2 SARS-CoV-2 experiments

Similar to the bacteria experiments, we perform 4 sets of experiments for SARS-CoV-2, using genomes downloaded from GISAID [14] (June 12, 2022). Sequences from the period 1 January 2021 until 31 March 2021, satisfying a minimum genome length of 25,000 nucleotides and an N-content of at most 0.1%, were randomly selected to a maximum of 1000 genomes per lineage. This is our baseline reference set. We performed a similar selection step with the additional constraint that genomes were sampled from a specific geographic region: the continent North America, the country United States of America, and the state Connecticut, resulting in a continent-filtered, country-filtered, and a state-filtered reference set, respectively. By simulating reads from SARS-CoV-2 genomes sourced from the state Connecticut, this setup allows us to assess the effect of filtering genomes in the reference database based on proximity to the sample.

We run the same selection tools as in the bacterial experiments; however, given the high similarity between SARS-CoV-2 genomes, we use different similarity thresholds. Furthermore, for these experiments we also include VSEARCH (v2.28.1) and VLQ's allele frequency-based selection methods [2]. For VSEARCH and Gclust we use similarity thresholds of 95%, 99% and 99.9%, while using similarity thresholds of 95% and 99% for MeShClust as 99% similarity is the maximum allowed. We use MASH with a sketch size of 5000 and a $k$-mer size of 31 in order to achieve higher sensitivity, since SARS-CoV-2 genomes are highly similar. We then ran GGRaSP with automated threshold selection. For the single-linkage and complete-linkage clustering selections, we calculated the 1, 5, 10, 25, 50, 90 and 99 percentile distances for every lineage and converted these to similarity score thresholds. Finally, we ran VLQ with a minimal allele frequency threshold of 50% which ensures that, given the considered genomes, point-wise mutations (compared to the MN908947.3 reference genome) that occur with a frequency of at least 50% are captured in the selected genomes.

The reference sets were processed following the VLQ pipeline, which uses kallisto [4] to estimate SARS-CoV-2 lineage abundances. In our experiments we used kallisto version 0.44.0 with default parameters for building the reference indexes, as well as performing the predictions. These predictions were further processed with the VLQ pipeline, which converts kallisto output to relative abundances at the lineage level.

For these experiments we simulated samples of paired-end Illumina HiSeq2500 reads from SARS-CoV-2 genomes that were uploaded to GISAID, having Connecticut as their location and April 30th, 2021, as their sampling date in the corresponding metadata. We simulated 20 samples of approximately 9,000-10,000 reads with a length of 150bp and a mean fragment size of 250bp (std 10bp) at relative abundances of lineage B.1.1.7 of 1%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100%. The sample background (remaining

abundance) was selected randomly from other lineages sampled on April 30th, 2021, in Connecticut.

## 2.3 Performance evaluation

All experiments presented in this work were run on the DAIC high performance computing cluster [6]. The command lines that were used to run the tools can be found on Github.

### 2.3.1 Reference set comparisons

In every experiment, we calculate the containment index (CI) between every pair of reference sets $(R_1, R_2)$ defined as $\mathrm{CI}(R_1, R_2) = \frac{|R_1 \cap R_2|}{|R_1|}$. This metric quantifies the similarity of reference sets. Additionally, we report the total number of genomes selected by every method. For computational resource consumption, we record the peak memory usage and runtime (measured as user time + system time) for every method in every experiment.

### 2.3.2 Taxonomic profiling evaluation

To evaluate the accuracy of taxonomic profiling, we calculate the L1-norm, F1-score relative to the groundtruth, and the percentage of unclassified reads for the profiling results of every method, sample and reference set. For Kraken2 in combination with Bracken (from hereon simply denoted Bracken) we assume that the output corresponds to sequence abundances (see Methods), whereas BWA in combination with DUDes (from hereon denoted as DUDes) yields taxonomic abundances. Although Centrifuge outputs both sequence and taxonomic abundances, we have chosen to consider the sequence abundances obtained from the uniquely mapped reads, as this seems to work best in all cases.

In addition to profiling accuracy, we also report the resource requirements (as in reference set comparisons) for building the profiling indexes and profiling the samples. Finally, we also provide the storage size of the profiling indexes.

# 3 Results

In this section, we present the results of our experiments, structured around two main scenarios: bacterial genomes and SARS-CoV-2 genomes. For each scenario, we compare the outcomes of genome selection methods, analyzing computational performance and the composition of the resulting reference sets. We then evaluate how these selections impact the performance of taxonomic profiling across different levels of similarity and classification resolution.

## 3.1 Bacteria experiments

### 3.1.1 Reference genome selection

We started by running the dereplication tools on all "Complete Genome" quality bacterial genomes downloaded from NCBI. While in future steps we will only use a subset of these selections, this gives us a realistic estimate of what can be expected when using these tools for reference genome selection in practice. Furthermore, since the sequences are selected per species, these results directly generate reference genome selections for the species of interest.

The results of this analysis are summarized in Table 2, which highlights significant differences in runtime and memory usage across methods. While Gclust requires up to 548 days of computation time at a similarity threshold of 99%, MASH in combination with hierarchical clustering requires only 6.5 hours and less than a gigabyte of memory. MeShClust runs considerably faster than Gclust, but this comes at the expense of increased memory requirements. We also observe that all MASH-based approaches (including GGRaSP) select substantially fewer genomes in total compared to MeShClust and Gclust, even when using the same similarity threshold. This may be a consequence of using a relatively small sketch size (1000). Finally, note that genome selection can easily be parallelized by processing each species independently, or using the multi-threading option provided by some dereplication tools.

Figure 2 shows the containment indices between all 15 reference sets, including the All reference set. We observe that MeShClust and Gclust find relatively similar reference sets, although this is primarily a consequence of both methods selecting a large number of genomes. This also results in a relatively high containment index between any of the smaller reference sets and a MeShClust or Gclust reference set, when compared to the containment index of the reverse.

### 3.1.2 Taxonomic profiling

For each experiment (genus, family, order), we applied the dereplication tools to generate reference sets containing all bacterial species from the corresponding taxon. For example, for the genus experiments, we included genomes from all species of the *Streptococcus* genus. For each reference set we built the corresponding reference indexes for Bracken, Centrifuge and DUDes. Supplementary Tables S6-8 show the computational resource requirements for building these indexes. We observe that, regardless of the profiling

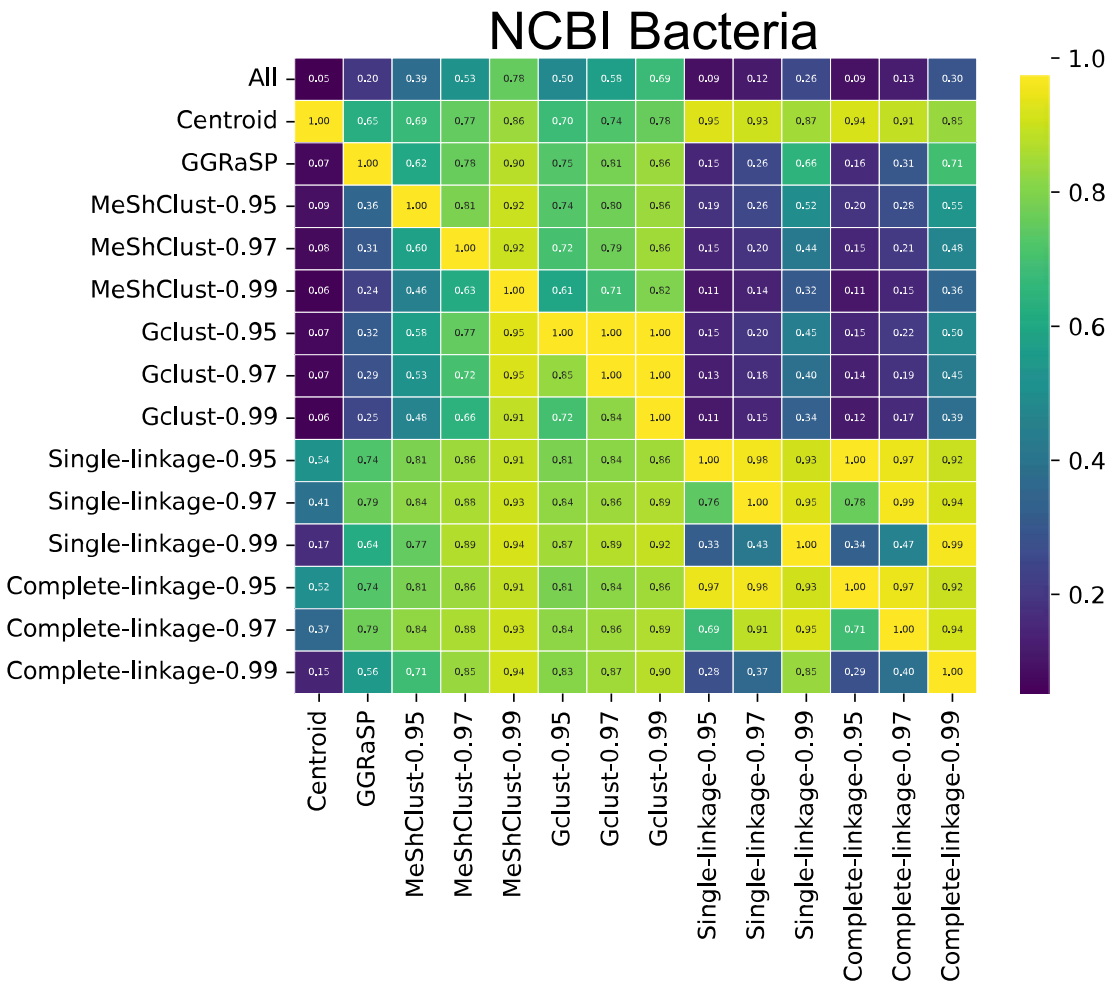


Figure 2: Containment indices between the reference genomes selected by different sequence dereplication tools. The containment indices are only calculated for species that have more than 1 genome (i.e. where selection can be applied), and if a tool failed to find a selection for a species, that species is excluded when calculating the index.

Table 2: Total runtime, peak memory usage and number of selected genomes after running the sequence dereplication tools on the Complete Genome quality bacterial NCBI Assembly genomes.

| Method | CPU time (h) | Mem (GB) | # genomes |
|---|---|---|---|
| All | - | - | 53536 |
| Centroid[1] | 6.1 | 0.65 | 11997 |
| GGRaSP[1,2] | 26.5 | 8.22 | 18987 |
| MeShClust-0.95 | 4979.7 | 158.97 | 26742 |
| MeShClust-0.97 | 4656.8 | 162.12 | 33003 |
| MeShClust-0.99 | 4328.0 | 160.45 | 43689 |
| Gclust-0.95[3] | - | 75.06 | 31476 |
| Gclust-0.97 | 10377.6 | 75.07 | 35210 |
| Gclust-0.99 | 13158.9 | 75.09 | 40137 |
| Single-linkage-0.95[1] | 6.5 | 0.74 | 13754 |
| Single-linkage-0.97[1] | 6.5 | 0.74 | 14905 |
| Single-linkage-0.99[1] | 6.5 | 0.74 | 21103 |
| Complete-linkage-0.95[1] | 6.2 | 0.74 | 13851 |
| Complete-linkage-0.97[1] | 6.2 | 0.74 | 15379 |
| Complete-linkage-0.99[1] | 6.2 | 0.74 | 22900 |

[1]Includes the runtime and memory usage for running MASH.
[2]GGRaSP failed to produce results for 1386 species, in which case a single representative was picked at random.
[3]The Gclust-0.95 runs failed to record CPU time.

method, there is a clear relation between the number of genomes and the resource requirements. Out of all three profiling methods, Bracken seems to work well for these experiments, requiring less than 30GB memory for constructing the All reference sets. Although the runtimes are high for all three tools, both Bracken and Centrifuge allow multi-threading to reduce wall-clock times (in contrast to BWA). Finally, we observe a clear benefit of a compressed genome representation: Bracken and Centrifuge require only 29-33 GB to store the All reference set, compared to the 122 GB required for the corresponding DUDes index.

The taxonomic profiling accuracy achieved by using the different reference sets is shown in Figure 3, which shows that the All reference set generally outperforms the other reference sets in all experiments. Although the difference between outcomes is often statistically significant, this is mainly a consequence of the low variance in outcomes. Considering this, it is clear that the MeShClust and Gclust reference sets with a similarity threshold of 97% or 99% are very competitive with the All reference set, achieving similar accuracies with all profiling methods, whilst requiring fewer computational resources. Interestingly, the MASH-based reference sets always get outperformed by the All reference set, with only the complete-linkage reference set at a threshold of 99% appearing to get close to the accuracy of the All reference set. Overall, there does not seem to be a clear trend in how outcomes change when moving from the genus experiments to the order experiments.

Regarding the resource consumption during profiling, we notice that both the memory usage and runtime of Bracken are relatively unaffected by the size of the reference index. Although the difference between the size of the smallest (centroid) and largest (All) reference index is nearly 25GB, the average difference in computation time and memory usage is less than 40 seconds and 1.5GB respectively. For Centrifuge and DUDes these differences are far more extreme, primarily because these methods load the whole reference index into memory during profiling.

Finally, there are substantial differences in the percentage of unclassified reads between reference sets. When using Bracken or DUDes, it appears that using a bigger reference set nearly always leads to classifying more reads. Most apparently, the centroid reference set achieves an average of 15.34% (13.78%) unclassified reads for Bracken (DUDes), compared to the other reference sets which have percentages unclassified between 5% and 7% (3%-8%). Interestingly, we observe the reverse for Centrifuge, where larger reference sets seem to consistently classify fewer reads.

## 3.2 SARS-CoV-2 experiments

### 3.2.1 Reference genome selection

We ran the sequence dereplication tools on the genomes of all lineages in the baseline, continent-filtered, country-filtered and state-filtered datasets resulting in 26 reference sets for every dataset. Supplementary Tables S2-5 shows the corresponding runtimes, memory usage and selected numbers of genomes. We observe that most methods finish running within three days of computation time, except for VSEARCH—this tool required up to 15.5 days to finish running (with a similarity threshold of 99.9%), since it performs exact alignment to calculate pairwise distances. Furthermore, VSEARCH and Gclust only select a small fraction of the 115850 genomes while using a similarity threshold of 99.9%, which highlights the fact that SARS-CoV-2 genome sequences of the same lineage are very similar.

Table 3: Number of genomes selected by every method for the order experiment, as well as the time and memory requirements for building the reference indexes, and the average percentage of unclassified reads. Bracken (both Kraken2+Bracken) indexes were built using 64 threads, Centrifuge indexes were built using 32 thread and DUDes (BWA+DUDes) indexes using 1 and 8 threads respectively.

| Method | Genomes | Index construction CPU time (h) | | | Unclassified | | |
|---|---|---|---|---|---|---|---|
| | | Bracken | Centrifuge | DUDes | Bracken | Centrifuge | DUDes |
| All | 14680 | 211.58 (28.01GB) | 204.71 (351.76GB) | 31.81 (104.52GB) | 5.44% | 19.78% | 3.86% |
| Centroid | 691 | 30.95 (9.51GB) | 1.50 (38.93GB) | 1.34 (4.36GB) | 15.34% | 12.92% | 13.78% |
| GGRaSP | 1649 | 48.11 (10.49GB) | 3.24 (74.16GB) | 5.85 (10.27GB) | 5.92% | 3.97% | 4.72% |
| MeShClust-0.95 | 3802 | 65.65 (10.98GB) | 30.38 (85.49GB) | 8.76 (26.03GB) | 5.67% | 9.12% | 4.37% |
| MeShClust-0.97 | 6419 | 102.37 (16.22GB) | 34.60 (324.27GB) | 13.61 (45.96GB) | 5.50% | 13.09% | 3.95% |
| MeShClust-0.99 | 11658 | 282.14 (23.00GB) | 120.49 (407.48GB) | 25.66 (83.11GB) | 5.45% | 19.23% | 3.88% |
| Gclust-0.95 | 6967 | 186.27 (15.49GB) | 66.43 (94.52GB) | 19.53 (49.44GB) | 5.55% | 16.68% | 6.29% |
| Gclust-0.97 | 8694 | 311.73 (18.32GB) | 122.33 (414.91GB) | 23.44 (62.00GB) | 5.54% | 17.42% | 6.28% |
| Gclust-0.99 | 10870 | 157.80 (21.99GB) | 212.09 (249.38GB) | 24.94 (77.81GB) | 5.45% | 18.16% | 3.85% |
| Single-linkage-0.95 | 870 | 42.36 (9.69GB) | 1.16 (36.48GB) | 1.53 (4.76GB) | 6.56% | 4.30% | 7.99% |
| Single-linkage-0.97 | 929 | 122.34 (9.81GB) | 1.27 (37.53GB) | 1.68 (5.15GB) | 6.41% | 4.19% | 5.70% |
| Single-linkage-0.99 | 1888 | 79.40 (10.52GB) | 6.27 (96.84GB) | 5.75 (11.89GB) | 5.69% | 4.56% | 4.27% |
| Complete-linkage-0.95 | 881 | 9.38 (9.62GB) | 1.15 (37.14GB) | 2.03 (4.84GB) | 6.53% | 9.95% | 7.94% |
| Complete-linkage-0.97 | 965 | 10.60 (9.98GB) | 1.47 (42.51GB) | 1.57 (5.39GB) | 6.26% | 7.60% | 5.45% |
| Complete-linkage-0.99 | 2746 | 83.24 (10.91GB) | 7.43 (141.96GB) | 5.51 (18.14GB) | 5.56% | 7.39% | 4.04% |

Figure 4 shows that there is only a substantial overlap in reference sets selected by the same method, except for VSEARCH and Gclust. These methods both use greedy incremental clustering, resulting in relatively similar reference sets and therefore large containment indices.

### 3.2.2 Lineage abundance estimations

Supplementary Tables S2-5 show the time required and peak memory usage for constructing the kallisto indexes needed for VLQ, as well as the average runtime and peak memory usage to profile the samples. We observe that constructing the reference index takes just over an hour for the All reference set, and smaller reference sets are indexed considerably faster. The memory usage increases significantly when more genomes are included, with the All reference index requiring more than 120GB of memory during construction.

Figure 5 shows the L1 distances and F1 scores of all methods (only including the best parameter setting for readability purposes) in all experimental settings. We observe that performance depends on the relative abundance of lineage B.1.1.7 in the sample. The complete-linkage based reference set achieves the best L1 score in all cases, while the centroid and single-linkage based reference achieve the best F1 score in most cases. Moreover, for the F1 scores we see that, except the state-filtered scenario, any reference set outperforms the All reference set since the latter is very susceptible to false positives.

When we compare the same method between scenarios (Supplementary Tables S13-18), we see that in nearly all cases abundance estimates get significantly more accurate (both in terms of L1 distance and F1 score) when a stricter proximity filter is applied. This shows that, in addition to achieving better results by applying sequence dereplication on the reference database, significant improvements can also be achieved by leveraging prior information about the sample.

Considering computational resources, Supplementary Tables S2-5 shows that using a larger reference set significantly increases computational resource requirements, with the runtimes per sample ranging from 16 seconds (centroid) to 40+ minutes (All). Furthermore, memory usage is also considerably higher for the larger reference sets, as kallisto needs to load the reference into memory when analyzing a sample. Overall, most well performing reference sets only require up to a minute of processing time while requiring up to 5GB of memory, which is substantially less than needed for the All reference set.

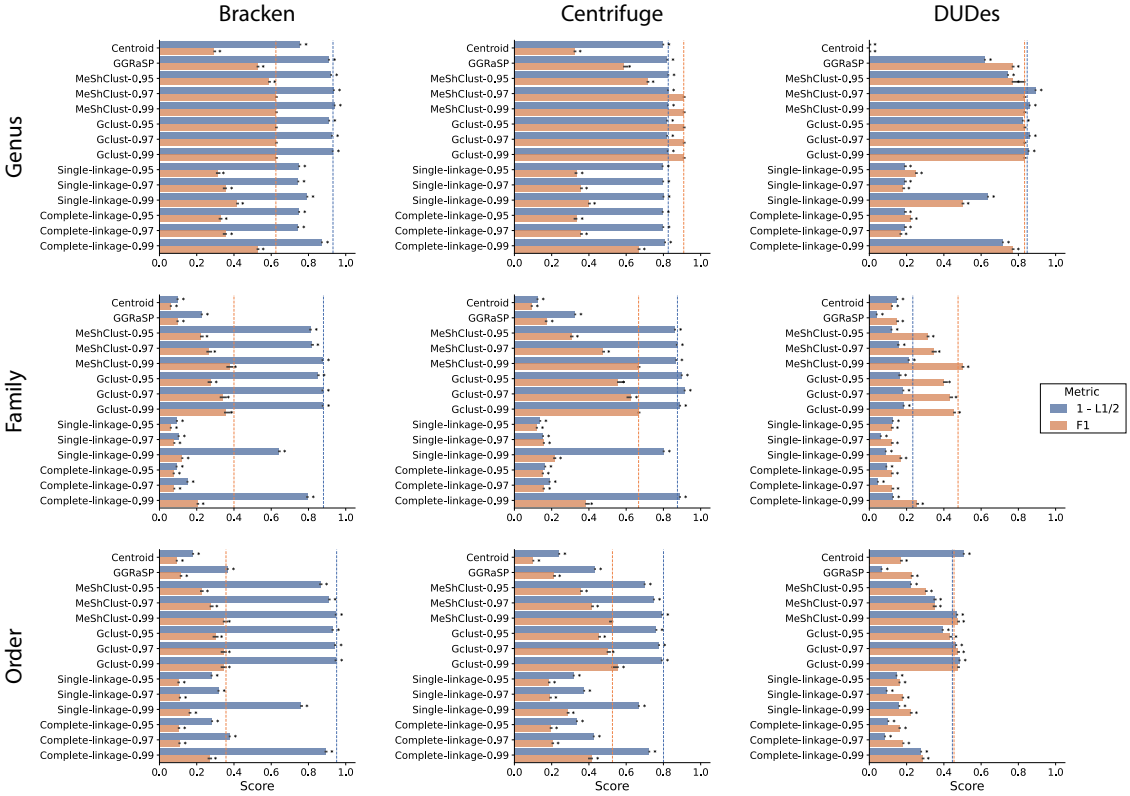


Figure 3: Accuracy metrics for the taxonomic profiling tools. Dashed vertical lines correspond to the median values of the All reference set, and an asterisk after a bar indicates that the outcome is significantly different from the all reference set (Wilcoxon signed rank test, $p < 0.05$).

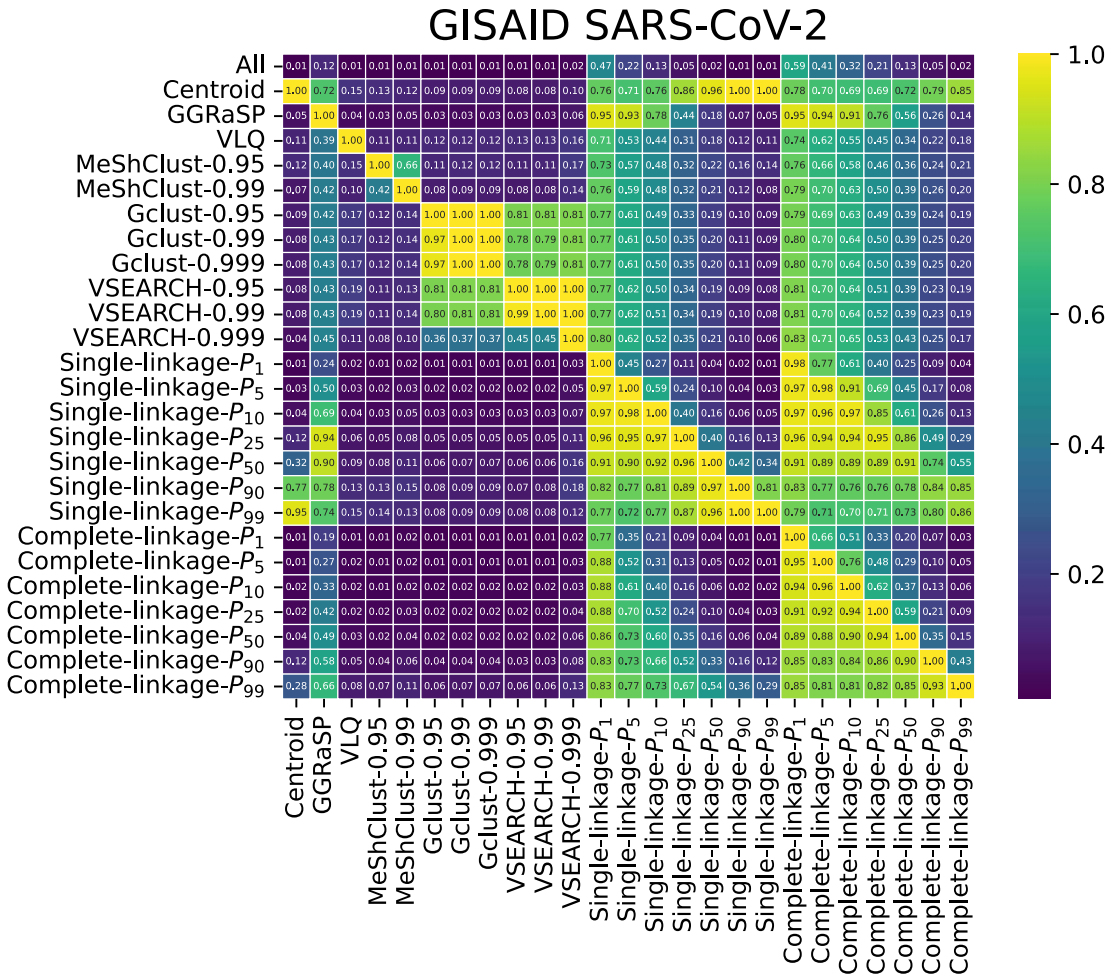


Figure 4: Containment indices between the reference genomes selected by different sequence dereplication tools. The containment indices are only calculated for lineages that have more than 1 genome (i.e. where selection can be applied), and if a tool failed to find a selection for a lineage, that lineage is excluded when calculating the index.

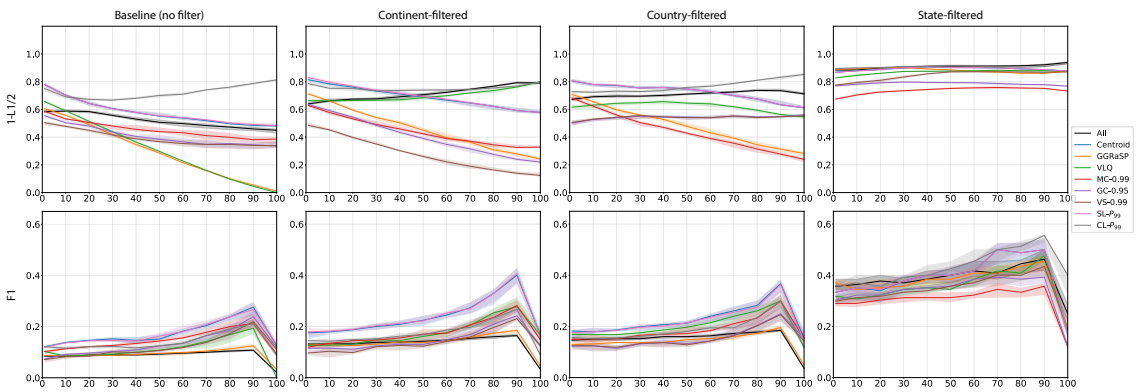


Figure 5: Accuracy metrics for the SARS-CoV-2 lineage abundance estimations. Only the parameter that achieved the best average score over all scenarios is shown for every selection method.

# References


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

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

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

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

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

[6] Delft AI Cluster (DAIC). The delft ai cluster (daic), rrid:scr_025091, 2024.

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

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

[9] P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *Proceedings 41st annual symposium on foundations of computer science*, pages 390–398. IEEE, 2000.

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

[11] H. 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.

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

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

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

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

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

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

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

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

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

[21] A. B. McIntyre, R. Ounit, et al. Comprehensive benchmarking and ensemble approaches for metagenomic classifiers. *Genome biology*, 18:1–19, 2017.

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

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

[24] M. R. Olm, C. 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.

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

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

[27] D. H. Parks, M. 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.

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

[29] V. C. Piro, M. S. Lindner, and B. Y. Renard. DUDes: a top-down taxonomic profiler for metagenomics. *Bioinformatics*, 32(15):2272–2280, 2016.

[30] K. D. Pruitt, T. Tatusova, and D. 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.

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

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

[33] H.-J. Ruscheweyh, A. Milanese, et al. mOTUs: profiling taxonomic composition, transcriptional activity and strain populations of microbial communities. *Current Protocols*, 1(8):e218, 2021.

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

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

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

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

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

[39] J. van Bemmelen, J. A. Baaijens, and I. Nika. Simulated reads and genome accession ids, Jan. 2025.

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

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

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

[43] D. E. Wood, J. Lu, and B. Langmead. Improved metagenomic analysis with kraken 2. *Genome biology*, 20:1–13, 2019.

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

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

# 4 Competing interests

No competing interest is declared.

# 5 Author contributions statement

J.B., J.A.B. and I.N. conceived the experiments, J.B. and I.N. conducted the experiments, J.B. analysed the results, J.B., J.A.B. and I.N. wrote and reviewed the manuscript.

# 6 Data availability

The scripts used to process the datasets in this paper are available on
Github (https://github.com/JaspervB-tud/reference_set_selection_benchmark/tree/main).

The datasets used here are available on Zenodo (https://doi.org/10.5281/zenodo.14727633)[39].

# 7 Acknowledgments

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.