VirStrain: a strain identification tool for RNA viruses

Herui Liao, Dehan Cai and Yanni Sun

Electrical Engineering Department, City University of Hong Kong, Kowloon, Hong Kong SAR

December 2020

1 Abstract

Genome epidemiology, which uses genomic data to analyze the source and spread of infectious diseases, provides important information beyond interview-based methods. Given fast accumulation of sequenced viral genomes, a basic need in genome epidemiology is to identify which reference genomes are identical or closest to the ones in a sequenced sample. Then the associated metadata such as the geographical locations can be utilized to infer the transmission network. In this work, we deliver VirStrain, a fast and accurate tool for conducting strain-level analysis from short reads. By using a greedy covering algorithm, we are able to derive unique k-mer combinations for highly similar reference genomes. VirStrain is able to detect the most possible strain and also multiple strains that may simultaneously infect the same host. We tested VirStrain on three types of RNA viruses whose reference genomes have different similarity distributions. For each types of virus, we assessed VirStrain across multiple benchmark datasets of different properties and complexity. The experimental results on both simulated and real sequencing data show that VirStrain outperforms other strain identification tools.

2 Introduction

Advancements of sequencing technologies opened us new avenues for tracking the spread of infectious diseases. The most recent example is COVID-19. It triggered extensive efforts in sequencing SARS-CoV-2 from a large number of patients. By end of August 2020, there are 74,976 and 22,124 complete sequences publicly available from the Global Initiative on Sharing All Influenza Data (GISAID) [31] and GenBank [4], respectively. Besides SARS-CoV-2, there are also a large number of sequenced genomes for other RNA viral pathogens. For example, 152,938 and 927,873 genomes have been deposited into publicly available

databases such as NCBI for Influenza and HIV infections. Compared to traditional methods that rely on incidence data and interview-based contact tracing, these genomes have provided important complementary information on analyzing the source, transmission, and spread of the diseases [9, 26, 13, 22, 12]. A fundamental step in applying genome epidemiology is to identify known strains that are identical or closest to those in a sequenced sample. Towards this goal, we will deliver a tool that can efficiently and accurately identify known strain for RNA viruses.

Strain As pointed out in [38], it is difficult to give a universal definition of microbial strain. Depending on the context, strain can refer to a variant of a given virus with unique and stable phenotypic characteristics under natural conditions [21], or a specific viral genome [38, 13, 17, 1, 14]. In this context, strain refers to a specific viral genome.

RNA viruses usually lack strict proofreading mechanisms during replication, leading to new copies containing genetic variations from the parent strains. Thus, it is not rare that genetic variations are found in genomes sequenced from different patients. Many of these variations can be neutral and deleterious to the virus survival. However, some mutations are beneficial to the fitness of the virus. For some extensively studied viruses such as HIV and Influenza, there are known mutations that lead to a different immune response in host cells, drug sensitivity and resistance [20, 27].

For newly discovered viruses such as SARS-CoV-2, whether or not the identified genetic changes affect the strains' virulence and transmissibility needs substantial evidence. But the fast accumulation of genomic data sequenced from different patients worldwide has provided precious information for precision epidemiology for infectious disease control [22]. With associated metadata such as symptoms, geographical location, and travel history, the reference strain database can provide important information for inferring the transmission path when the contact history is ambiguous or missing.

Some studies that used the strain genomes for studying the spread of COVID-19 already showed that the clusters of the reference genomes are highly structured and are consistent with their geographical distributions [12]. Before COVID-19, analysis of strain genomes also made important contributions to precision epidemiology by local HIV breakout monitoring in Canada [28], Zika virus transmission study in Florida USA [11], and Ebola virus transmission in West Africa [25].

Given the development of NGS, sequencing the virus using either targeted or viral metagenomic sequencing methods from a large number of patients will become the norm during pandemics. Given short reads as input, we will output the reference genomes that are closest to the ones in the underlying sample. In addition, if subtype information is associated with the reference strain, users can also obtain the subtype, clade, or subclade of the virus.

The difficulty of strain identification depends a lot on the similarity between sequenced strains. Intensive sequencing efforts triggered by COVID-19 have

resulted in a large number of genomes with high sequence similarity partially because of the recent association of this virus with human and also its relatively low mutation rate. Short reads tend to be ambiguously mapped or aligned to multiple reference genomes. Dissolving the ambiguity in the alignment is computationally expensive [1]. Faster methods based on genomic-specific k-mers often cannot reach the strain level resolution [35, 7]. In addition, the strain identification tool should be able to detect more than one reference strain if there are multi-strain infection, which is not rare for RNA viral diseases.

Although there are some strain-level analysis tools for bacteria, not all can be repurposed to take virus genomes as inputs. Some other tools' cannot scale to the large number of strain genomes with high similarities. We summarize related work in the following section.

2.1 Related work

Popular taxonomic classification tools such as Kraken series [35, 5] are not able to conduct strain-level analysis when the reference genomes share high similarities. Our experimental results have shown that their performance on SARS-CoV-2 and Influenza is not satisfactory.

When near-complete virus genomes are available, alignment-based tools such as BLAST can be applied to find reference genomes with the highest similarity. If the inputs are short contigs or reads, alignment-based tools tend to align them to multiple genomes with the same scores. Similarly, other tools and websites that can monitor the mutations in strain genomes such as NextFlu and NextStrain [26, 13] also take genomes as input and assign the genome into major genetic groups. When there are difficulties in constructing high-quality virus genomes due to complexity of the data (e.g. metagenomic data), the low abundance of the virus, or the presence of a minor strain besides a major one, there is a need for a tool that can still identify the strains using reads as input.

There are some methods that focus on conducting strain-level analysis using short reads as input. Automated strain-level epidemiology analysis tools mainly focused on bacteria in microbiome sequencing data. The available reference-based strain-level analysis tools were divided into four groups by Yan et al. [38]. The first group focused on identifying known genotypes from reference genomes [15, 1]. The second group will further identify potentially novel strain of one bacterial species [33]. The third group of tools can identify one or multiple strains using reference-based methods [23, 29]. The last group uses structural variants rather than just single nucleotide variations (SNVs) [30], which usually renders better accuracy.

As the main goal of this work is to develop a method that can identify the closest strain to a known strain from short reads data, it is more related to the first group of the strain-level analysis tools for bacteria. The representative tools in the first group, PathoScope [15] and Sigma [1], rely on ambiguity-resolved read mapping strategies between short reads and reference genomes with high sequence similarity. Both tools allow users to create their own reference database and thus can be applied to viruses. However, they are too slow for identifying

strains with tens of thousands reference genomes and large-scale sequencing data.

Other bacteria-centered tools cannot be conveniently re-purposed for virus strain analysis because they use bacteria-specific features such as bacterial marker genes, structural variants etc. For example, our experiments showed that existing marker gene sets can recognize HIV if the reads are sequenced from the dominant strain HXB2, but not other strains of HIV [6]. One possible reason is that the derivation of the marker genes by existing programs [33] did not use all the available viral strains and thus can miss many strains.

In addition, RNA viruses usually possess higher mutation rates than bacteria because of the error-prone replication process. As a result, different patients contain different viral genomes. In particular, for a new breakout such as COVID-19, a large number of sequenced strains within a short period often share extremely high sequence similarity, making identifying individual strains from new data more challenging. Use the available haplotypes or strains to infer transmission has been applied to COVID-19. For example, Gudbjartsson et al. [12] are able to assign a sample to its closest haplotype based on a manually derived haplotype table. However, it is not clear the manually created table can scale to larger datasets or other viruses.

A very relevant work with the similar challenge is QuantTB [3], which targeted at identifying individual *M. tuberculosis* strains with high similarity. However, their tool is "hard coded" for *M. tuberculosis* and thus we cannot conveniently extend it to viruses. In addition, they also applied different thresholds on the number of distinct SNPs between strains, which are actually still stringent for newly identified RNA viruses such as SARS-CoV-2

3 Methods

In this work, we developed a tool, VirStrain, which can quickly identify one or multiple reference strains closest to those in short-read sequencing data. It achieves a better trade-off between speed and resolution by deriving k-mer sets that can distinguish highly similar strains. VirStrain conducts strain identification using short reads as input and doesn't rely on sequence assembly, making it more amenable to cases where full virus genome cannot be assembled. The output of VirStrain contains the most possible strains identified in the data and the detailed read coverage of its single nucleotide variation (SNV) sites in an interactive HTML format.

Highly similar reference genomes may not possess genome-specific k-mers. But they can possess genome-specific k-mer set, where all the component k-mers can be utilized together to distinguish different reference genomes. Fig. 1 shows a toy example of using k-mer sets to distinguish five sequences when there are no genome-specific k-mers. In order to find such k-mer set, we develop a greedy covering algorithm to identify unique combinations of SNV sites from aligned virus reference genomes. Then k-mers will be extracted from the SNV sites and construct k-mer set for underlying genomes.

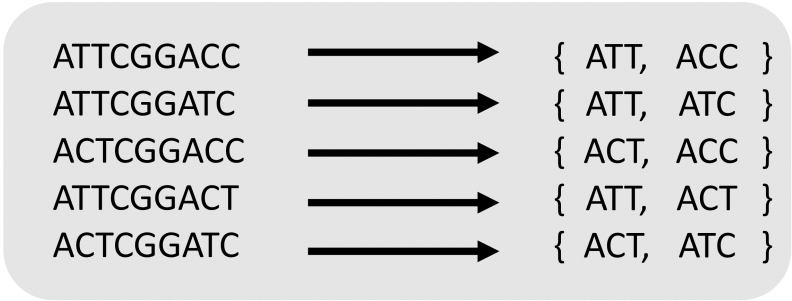

Figure 1: Use k-mer sets to distinguish five sequences of high sequence similarity. Each sequence has a unique k-mer combination.

3.1 Step 1: identify unique set of SNVs from reference genomes

The input to this algorithm is an MSA of the reference genomes. It is noteworthy that generating MSA for thousands to tens of thousands of genomes can be slow. But when the reference genomes share high sequence similarity (such as for SARS-CoV-2), the MSA can be produced using more efficient programs, such as the one provided by Mafft at its website [18].

Given the MSA M, we will exclude all the sites where no SNV is observed. Instead, we favor variations from conserved sites, which indicate features that are specific to one or a small number of genomes. Thus, given M, we compute the Shannon entropy $\mathcal{H}$ for each column. Let p^b be the probability of a nucleotide base b. $\mathcal{H} = \sum_{b \in \Sigma} -p^b log p^b$, where p^b is the percentage of a base b in the base set Σ. The column containing all identical bases has entropy $\mathcal{H} = 0$ while a column containing an identical percentage of bases has maximum entropy. Then we pick the column with minimum positive $\mathcal{H}$ from M. Let this column be s_i. Let the SNV event corresponding to nucleotide base b with the minimum p^b be s_i^{min}. All the genomes containing s_i^{min} at site s_i will be extracted and saved in one cluster. The entropy for the remaining genomes will be updated after the extraction. And this greedy choice will be applied to the remaining genomes until all the genomes are in one cluster. It should be noted that low-quality columns with too many dashes will not be considered and can be filtered in pre-processing. Depending on the reference genome similarity and alignment quality, users can choose a threshold for the allowed percentage of dashes in one column. For SARS-CoV-2 and H1N1(HA), our default cutoff is 0. For HIV, our default cutoff is 10%. The pseudocode of the entropy-based greedy covering algorithm is presented in Algorithm 1. A working example is shown in the top-left panel of Figure 2.

After we apply this greedy covering algorithm, the reference genomes are divided into multiple clusters, where each cluster is defined by one SNV event.

Algorithm 1 Divide reference genomes into clusters with unique SNV combinations using greedy covering algorithm.

Require: Multiple sequence alignment with multi-fasta format (such as the output by mafft). Let the alignment be M with M_i^j be the character at row j and column i.

M has n columns and m sequences.
$\Sigma = \{a, c, g, t, -\}$
Define an empty list S
Define a two-dimensional array N for recording the character occurrence times in each column
Initialize H as an empty list ▷ Initialization
for $i = 1$ to n **do**
    Compute N_i^b, the occurrence of the character b for $b \in \Sigma$. ▷ Some columns may contain N
    Define $p^b = \frac{N_i^b}{\sum_{b \in A} N_i^b}$ for $b \in \Sigma$
    Entropy $h = \sum_{b \in \Sigma} -p^b log p^b$
    if $h > 0$ **then**
        push(S, i)
        push(H, h)
    ▷ Only columns with entropy > 0 are saved
Define $nseq = 0$ ▷ nseq is the number of clustered sequences
while S != empty and $nseq < m$ **do**
    $n_{min} = argmin_i\ (H_i)$
    $b_{min} = argmin_{b \in A}\ N_{S[i]}^b$
    Initialize a new cluster C
    for $x = 1$ to m **do**
        if $M_{S[i]}^x$ eq c_{min} **then**
            push(C, x)
            if x has not been clustered **then**
                nseq++
    Output the new cluster C
    Remove n_{min} from S
    for $k = 1$ to $|S|$ **do**
        for $l = 1$ to $|C|$ **do**
            Let $b' = M_{S[k]}^{C[l]}$
            $N_{S[k]}^{b'} - -$
        update H_k

Figure 3 sketches the SNV sites for different clusters based on the order of SNV site selection in the greedy covering algorithm. Let the number of chosen SNV sites (i.e. the final number of clusters) be m. Let s_i be the SNV site chosen at the ith step in the greedy covering algorithm. Again, s_i^{min} is the base with minimum frequency at site s_i in the remaining genomes. Let the corresponding cluster be c_i, which contains the genomes containing s_i^{min} at the ith step. We use s_i^{min} at SNV sites s_1, s_2, ..., s_i to represent cluster c_i. We have the following theorem and proof.

Theorem 1. *Let the SNV base combination for the ith cluster c_i be $s_1^{min}, s_2^{min}, \ldots, s_i^{min}$. This nucleotide base combination uniquely represent the genomes in cluster c_i.*

Proof. In order to prove that the nucleotide base combination $s_1^{min}, s_2^{min}, \ldots, s_i^{min}$ uniquely represent the genomes in cluster c_i, we need to show that at least one base of these SNV events in c_i is different from any other cluster c_j, where $j \neq i$. Figure 3 can be used to illustrate this proof. Without losing generality, we consider two cases. In case 1, we consider a cluster c_j with $j < i$. The bases at j in genomes of c_j must be identical (condition for clustering). In addition, that base must be different from the base s_j^{min} in genomes of cluster c_i. Otherwise, the genomes of c_i will be clustered into c_j at step j. In the second case, we consider a cluster c_j with $j > i$. For any genomes in c_j, their base at SNV site s_i must be different from the base s_i^{min} of the genomes in c_i. Otherwise, that genome will be clustered into c_i. Thus, we proved that at least one base at the SNV event combination in c_i is different from any other cluster c_j.

Thus, as shown by Figure 3, we identified unique SNV combinations for each cluster. As we found the columns involving indels may have alignment errors or assembly errors (an example can be found in Supplementary Figure S4), we only use columns with no gap or a small number of gaps in greedy covering. As a result, some clusters can contain multiple genomes. In this case, the genomes inside each cluster can be aligned again (to reduce the alignment errors) and be clustered in a hierarchical fashion. An example can be found in the bottom panel of Figure 2.

3.2 Step 2: *k*-mer extraction from SNV sites

We will extract k-mers from these SNV sites, with the center base of each k-mer coming from this site. The complete k-mer is then constructed by including $\frac{k-1}{2}$ bases to the left and right side of a variation site. Figure S3 in the supplementary material shows an example of k-mer extraction.

If k-mers repeat at different sites in the reference genomes, false positive strain identification will happen. Intuitively, bigger k-mers tend to appear less frequently than small k-mers. We thus investigate how many times we will observe the same k-mer at different sites in an MSA. As the reference genomes SARS-CoV-2 have high sequence similarity and larger genome sizes than other known RNA viruses, we conducted this experiment on SARS-CoV-2. The results

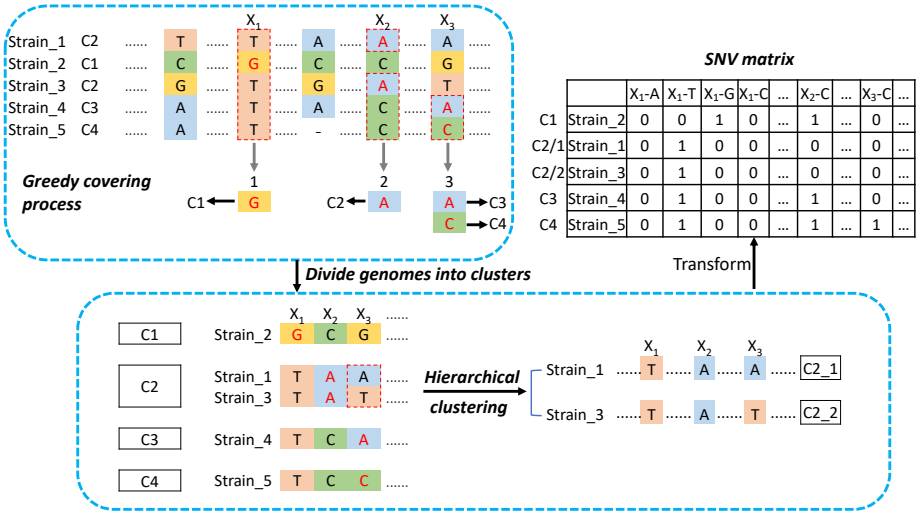

Figure 2: The sketch of the k-mer identification and SNV matrix construction stages. The SNVs in the matrix are represented by the site and base [3]. "C1-C5" represents the name of clusters.

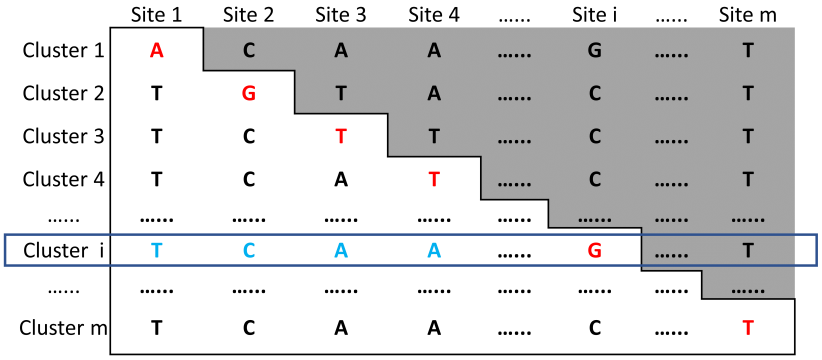

Figure 3: The SNV sites and corresponding clusters. The base in red denotes the base with the minimum p^b (i.e. s_i^{min}). Genomes in the same cluster share the red base at the chosen SNV site. Each cluster has unique SNV base combination, which is shown in the white part of each row. The SNV base combination for cluster c_i is shown in a box.

are summarized in Supplementary Figure S2. As the figure shows, with the increase of k, the repeat numbers of the k-mer at different sites reduce quickly. Thus, either large k can be chosen or we can directly exclude those columns that contain repetitive k-mers in order to fulfill the assumption.

3.3 Iterative strain search algorithm

After we identify the k-mer combination for each cluster, we will conduct strain search following the strategy from QuantTB [3].

In the ideal case of no sequencing errors and each base of a viral genome being covered by at least one read, the k-mers extracted from the SNV site set that uniquely represent genomes in c_i are sufficient to determine the group or the strain precisely. But in reality, both heterogeneous coverage and sequencing errors exist. For clusters represented by a small number of SNV sites (e.g. c_1 contains just one SNV site), sequencing errors can reduce both the precision and recall. Thus, we will use all the k-mers corresponding to all m SNV sites for strain identification. If the original SNV sites can distinguish the genomes in different clusters, adding SNV sites will not change this property. Thus, each cluster still possess unique k-mer combinations.

To detect all possible strains, we take an iterative strategy similar to QuantTB [3]. The overall workflow of the strain search algorithm is displayed in Figure 4. For an input set of reads, the k-mer match frequencies are computed using a k-mer counting tool and are mapped to an SNV matrix, which will allow us to quickly compute the sum of the coverage for all the SNV sites and rank the strains. The major operations are described below.

Construction of the SNV matrix The SNV sites chosen by the greedy covering algorithm will be used to construct an SNV matrix S of size $4mn$, where m is the number of chosen variation sites and n is the number of reference genomes. An example is given in Fig. 2. For a strain i and a chosen SNV site x, there are four cells corresponding to bases A, C, G, and T in S. Denote an SNV event as $x - b$, indicating that base b is observed at site x. A cell $S_{i,x-b}$ is 1 if the strain i has base b at site x. Otherwise, it is 0. Each cell in the matrix S has associated k-mers match frequency.

Rank the reference genomes using *k*-mer match frequency We apply Jellyfish (V2.3.0) [24], a fast multi-threaded k-mer counter, to count k-mers in the sequencing data. Let F_{x-b} be the k-mer match numbers of base b at site x. Thus, $S_{i,x-b} = S_{i,x-b} * F_{x-b}$. Then, we will compute the frequency of base b at site x by normalize the F_{x-b}. Therefore, $S_{i,x-b} = \frac{S_{i,x-b}}{\sum_{b\in\Sigma} S_{i,x-b}}$. To reduce the effect of sequencing error, we filter $S_{i,x-b}$ if its value is smaller than a given threshold.

Once S is updated based on the actual k-mer match frequency from the reads, we will compute the score of strain i using $vscore_i = \frac{\sum_{x=1\ldots m} S_{i,x-b}}{\sum_{x=1\ldots m} I(S_{i,x-b}>0)}$, where I is an identity function. $vscore$ favors strains with the most number of k-mer hits. Although it looks reasonable to consider other factors such as uniformity of k-mer match frequency, our empirical studies show that considering the total number of k-mer hits renders the best accuracy. One possible reason is the heterogeneous coverage of real sequencing data along RNA viral genomes.

Read coverage profiles of 11 real sequencing datasets in our experiment can be found in Supplementary Figure S6. We will compute $vscore_i$ for all the strains and rank them in decreasing order.

Iterative strain search VirStrain will output the top 1 strain in the ranked list and then update S by replacing the frequency of all the $x - b$ pairs of the identified strain with 0. Any strains that share the same SNV base with the identified strain cannot reuse the frequency. Otherwise, strains that share high similarity with the identified one can be easily selected as false positives. Then the remaining SNV events will be used in the next iteration to identify other possible strains. At each iteration, the sequencing depth is calculated by taking the average frequency of its SNVs for each identified strain in the sample.

VirStrain continues to calculate the score and identifies the most possible strain in each iteration until the frequency values of all variations become 0. In the end, this iterative process will return a list of strains with their k-mer coverage profiles.

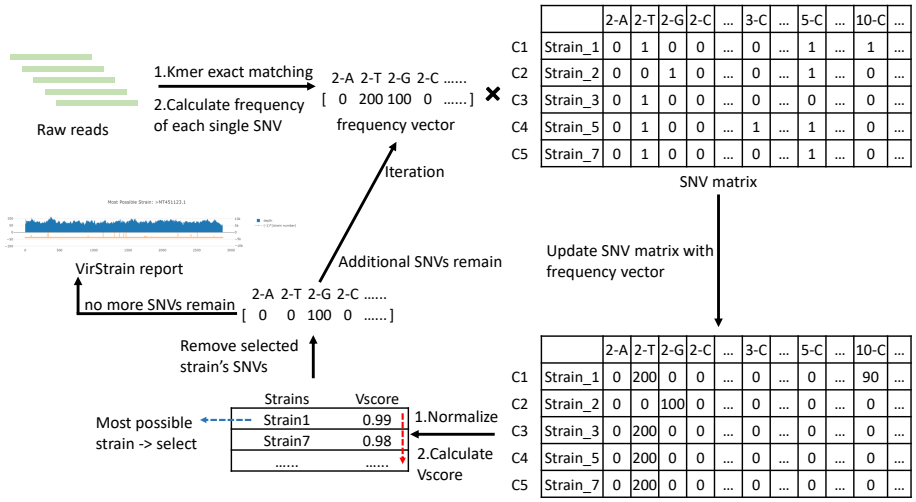

Figure 4: The overall workflow of the iterative strain identification. The red arrow represents all strains are in descending order of $Vscore$. The usage of the SNV matrix is inspired by QuantTB.

3.4 Output format

The output of VirStrain contains two files. The first is a report file in text format. This file contains all identified strains and their depth and site coverage, etc. The other file is an interactive HTML page to display the depth and uniqueness of sites. An example is shown in Supplementary Figure S5.

4 Results

In order to consider different cases, we tested our tool on three types of RNA viruses with different mutation rates and strain-level similarity. The first is SARS-CoV-2, many of which have very high sequence similarity and may differ only at a few sites. The second is the "HA" region of Influenza A H1N1, which has lower average similarity than SARS-CoV-2 but higher similarity within the same clades and sub-clades. The third is HIV, which has a much lower similarity than SARS-CoV-2 and H1N1. For HIV, we used the "Gag" region, which is one of the marker genes for HIV subtype classification [32]. Figure 5 compares the pairwise Hamming distance at the SNV sites chosen by our algorithm from the aligned reference sequences.

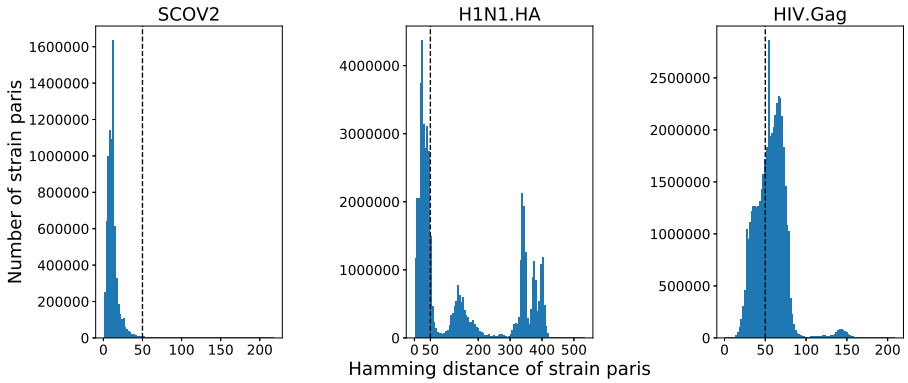

Figure 5: The pairwise Hamming distance distribution of SARS-CoV-2, H1N1 (HA), and HIV (Gag). The Hamming distance here is measured only on the chosen SNV sites by the greedy covering algorithm. They can represent the overall similarity of the three reference sets. The dashed line is at distance value 50 across the three panels for easy comparison.

4.1 Overview of the data and clustering results

To construct the reference database for VirStrain, we collected all available complete genomes of SARS-CoV-2 from NCBI, H1N1 (HA) from Influenza Research Database (IRD, http://www.fludb.org), and HIV (Gag) from HIV database (http://www.hiv.lanl.gov) as of July 14, 2020. Table 1 summarizes the statistics of the reference genomes/genes of these three viruses and also their final clustering results. It shows that the number of HIV (Gag) sequences is nearly equal to the number of clusters while the other 2 viruses differ a lot, which is caused by relatively low sequence similarity between HIV (Gag). The cluster size distribution of three viruses are displayed in Figure 6. As shown in Figure 6, most clusters are very small, with many containing a single genome. But there are also genomes that cannot be distinguished by the chosen SNV sites.

Virus	Length Range	# Genome Number	# Cluster Number
SARS-CoV-2	(29432, 30018)	6872	4082
H1N1 (HA)	(1692, 1864)	11424	10272
HIV (Gag)	(1364, 1596)	9552	9545

Table 1: The statistics of the reference database of 3 viruses. "# Genome Number": the number of downloaded genomes. "# Cluster Number": the number of final clusters generated by VirStrain.

Especially, there are strains with main differences in the beginning and ending regions (such as different lengths). As a result, the beginning and ending parts of the alignment contain many gaps, making those columns not usable by our greedy covering algorithm. Thus, these strains are usually in the same cluster.

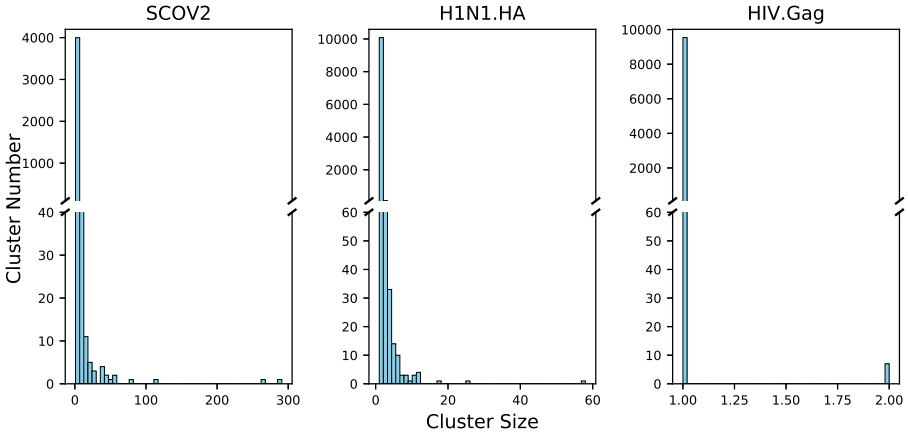

Figure 6: The cluster size distribution of SARS-CoV-2 , H1N1 (HA), and HIV (Gag). "Cluster size" represents the number of strains contained in each cluster.

4.2 Overview of the experiments

As there are already a number of successful higher-level taxonomic classification tools, our focus is thus to test whether VirStrain can recover the correct strain from reads that are sequenced from the virus of interest.

We demonstrated the performance of VirStrain in six different settings involving both real and simulated reads. We compared the accuracy of VirStrain with four popular strain-level identification tools, KrakenUniq [5], Pathoscope2 [15], Sigma [1], and Centrifuge [19]. In addition, we also tested Kraken2 [36]. Although Kraken2 is not designed for distinguishing highly similar strains, its

method of using all k-mers for classification can be compared with VirStrain that uses only selected k-mers from our greedy covering algorithm.

In the first experiment, we focused on testing VirStrain on detecting a single strain from simulated short reads when this strain is identical to one of the reference genomes. In the second experiment, we tested VirStrain on simulated two-strain data, where the major strain has 100x coverage and the minor one has 10x coverage. In the third experiment, we simulated novel SARS-CoV-2 strains that have 5 to 13 mutations from reference strains and evaluated whether the VirStrain and other tested tools can detect the closest reference genome. In the fourth experiment, we tested VirStrain on multiple real SARS-CoV-2 sequencing data. In the fifth experiment, we reported possible two-strain infection cases from SARS-CoV-2 sequencing data of Washington State of USA. Finally we tested VirStrain on a mock 5-strain HIV dataset.

We use accuracy as the main performance metric for different tools. It is worth noting that the tested tools usually output multiple strains with associated ranking. If we know the the number of strains (e.g. x) in a sample, we will keep only the top x outputs of a tool. Then the **accuracy** is defined as the percentage of correctly identified strains of the total number of outputs. For example, when evaluating the performance of detecting one strain, we keep the output of the highest rank and assess whether it is correct. The accuracy can be computed for just one dataset or multiple datasets. For experiments that test these tools on multiple datasets, we report the aggregated accuracy. For example, if we tested these tools on 100 datasets, each of which contains one strain, the accuracy will be percentage of correctly predicted strains in 100 outputs for each dataset.

4.3 Detecting the most possible strain from simulated reads

In this experiment, we compared VirStrain against Kraken2 [36], KrakenUniq [5], Pathoscope2 [15], Sigma [1], and Centrifuge [19]. Although there are more taxonomic classification tools for sequence classification, other authors have shown that they cannot achieve satisfactory performance on strain-level composition [8]. Thus, we did not include those tools in our comparison.

For each tool, the reference database is constructed using RNA viral strains. As Sigma is computationally expensive, we were not able to construct its database using all strains. To ensure a fair comparison using the same reference database, we built a smaller, lower-resolution database of 200 stains randomly selected from all strains of the three types of viruses. A larger-scale experiment for fast running tools will be described in the next section.

For each virus, we randomly picked 100 strains/genes from the 200 reference sequences and simulated short reads from each. Thus, there are 300 datasets for this experiment. For each dataset, we used ART [16] to simulate 250 bp error-containing Illumina reads with depth of 100X, average insert size of 600 bp, and standard deviation of 150 bp. We identified strains from these simulated reads with VirStrain and five other tested programs and calculated the accuracy for each program.

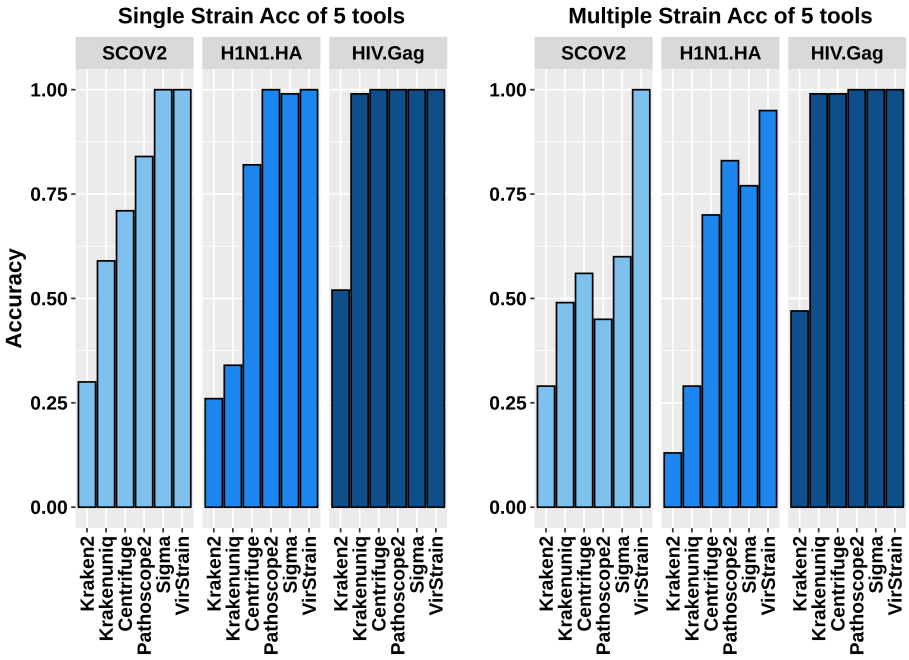

Figure 7: The accuracy (Acc) comparison of 6 tools. There are 100 sets of simulated reads for single-strain datasets and 100 for multi-strain datasets. For each set of multi-strain simulated reads, there are two strains with 100X and 10X coverage, respectively.

The performance comparison of different tools is shown in the left panel of Figure 7. Gag region of HIV shares relatively low similarity and thus it is easier to distinguish different reference genes. As a result, all tools have high accuracy. As H1N1.HA has very high similarity within the same clades or sub-clades, sequence classification tools that are not specifically designed for distinguishing highly similar genomes have low accuracy. We have similar observations for SARS-CoV-2 too. Across all the three viruses, VirStrain has consistently high accuracy.

VirStrain extracted about 30,000 k-mers out of roughly 300,000 k-mers from the input reference genomes. As shown in Figure 7, using all k-mers in Kraken2 does not render satisfactory accuracy. Similarly, we observed decreased accuracy if we use all possible k-mers in VirStrain. Thus, using greedy covering algorithm to select unique k-mer combinations is important to VirStrain.

4.4 Leave-one-out experiment

In order to test the performance of VirStrain on all strains, we carried out a leave-one-out experiment with VirStrain, KrakenUniq, and Centrifuge. As Sigma and Pathoscope2 are more computationally expensive than other tools, they are not compared in this experiment. Kraken2 is also not compared due to its low accuracy on the single-strain dataset.

In this experiment, one of the SARS-CoV-2 strains in the reference database was randomly selected as the test strain. We simulated reads from it and used the reads as input to the three tools. This process is repeated for all other strains. The accuracy is computed for all the strains and shown in Figure 8. VirStrain is able to identify all strains in the database while other tools have lower accuracy.

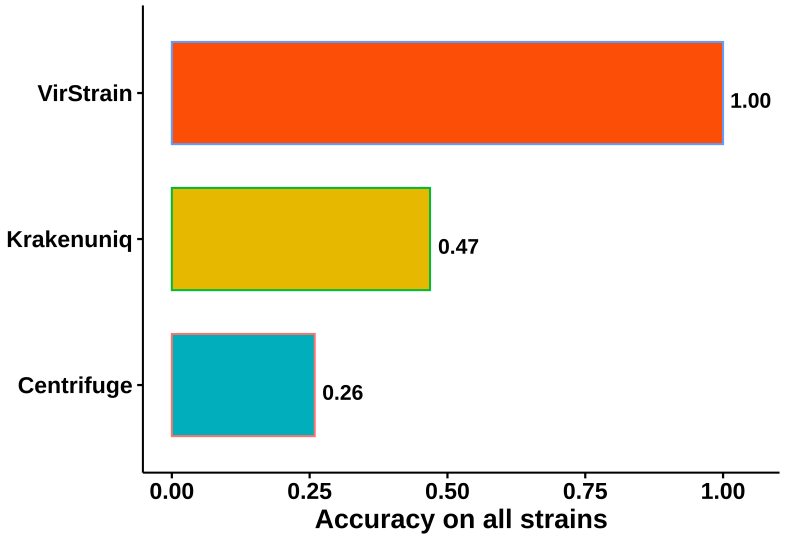

Figure 8: The accuracy comparison of 3 tools on all strains in the database. These three tools have fast speed and also reasonable accuracy in the small scale experiment on 200 reference genomes.

4.5 Detecting multiple strains from simulated data

Multi-strain infection is not rare for RNA viruses, especially the ones with high mutation rate such as HIV. Usually, if one strain dominates the virus population, the minor strains tend to be missed. To mimic this situation, we constructed two-strain datasets that consists of a major strain (100x coverage) and a minor strain (10x coverage). Similar to "single-strain" datasets, we constructed 100 datasets of simulated reads for each type of virus. Each set contains simulated reads from two randomly selected reference sequences. The read simulation

process is the same as the single strain experiment. As we know there are two strains, the accuracy of a tool is defined as the ratio of correctly identified strains to the total number of the two most possible strains output by each tool. The performance comparison is shown in the right panel of Figure 7. Although the accuracy of VirStrain decreases a little for H1N1.HA (from 1.0 to 0.95) compared to the single-strain experiment, it maintains the accuracy of 1.00 for SARS-CoV-2. And it outperforms other tools by about 10% on H1N1(HA) and 38% on SARS-CoV-2.

4.6 Running time comparison

To evaluate the computational efficiency of VirStrain, we compared the running time of the tested tools on the simulated data in Table 2. VirStrain has similar running time to Centrifuge and KrakenUniq but runs significantly faster than Pathoscope2 and Sigma. All the experiments were tested on an HPCC CentOS 6.8 node with 2.4Ghz 14-core Intel Xeon E5-2680v4 CPUs and 128 GB memory. We used 8 threads for all tools.

Data sets	VirStrain	KrakenUniq	Centrifuge	Pathoscope2	Sigma
Sim_single_strain (6 MB)	16s	8s	6s	110s	255s
Sim_multiple_strain (6.6 MB)	23s	9s	8s	140s	268s
SRR10971381 (19.5 GB)	215s (Y)	254s (N)	290s (N)	1721s (Y)	>15h (-)

Table 2: Running time of five tested tools on simulated and real data. Sim_single_strain and Sim_multiple_strain represent simulated single-strain and multiple-strain datasets, respectively. For real data, the identification result is represented by Y and N, where Y means correct identification and N means wrong identification. Sigma doesn't have the identification result due to its long running time.

These results indicate that VirStrain achieved much higher accuracy than those computationally efficient tools such as KrakenUniq and Centrifuge while maintaining comparable speed. It also outperforms those mapping-based tools like Sigma and Pathoscope2 on both accuracy and speed.

Relative abundance computation For identified strains, VirStrain also outputs its sequencing coverage, which can be used to compute relative abundance for multi-strain infection. As Figure 7 shows, accuracy of Kraken2, KrakenUniq, and Pathoscope2 on the SARS-CoV-2 multiple-strain data sets is lower than 0.5. Thus, we didn't include them in the comparison. Sigma and Centrifuge were able to return the strains' abundances in the outputs. Therefore it is convenient to calculate the relative abundance for each strain.

Figure 9 shows that the relative abundance estimated by VirStrain is closer to the ground truth than others. Sigma and Centrifuge failed to detect the minor strain in many datasets. Thus, many data points are aligned with x-value 0.00. In addition, they have more variations about the relative abundance

computation for different datasets even though the ground truth keeps the same (100x vs 10x). These results indicate that, even for mapping-based tools such as Sigma, it's very difficult to return accurate relative abundance estimation due to the high sequence similarity between SARS-CoV-2 strains. But VirStrain still outputs reasonable estimation of the relative abundance distributions despite the high similarity between the reference genomes.

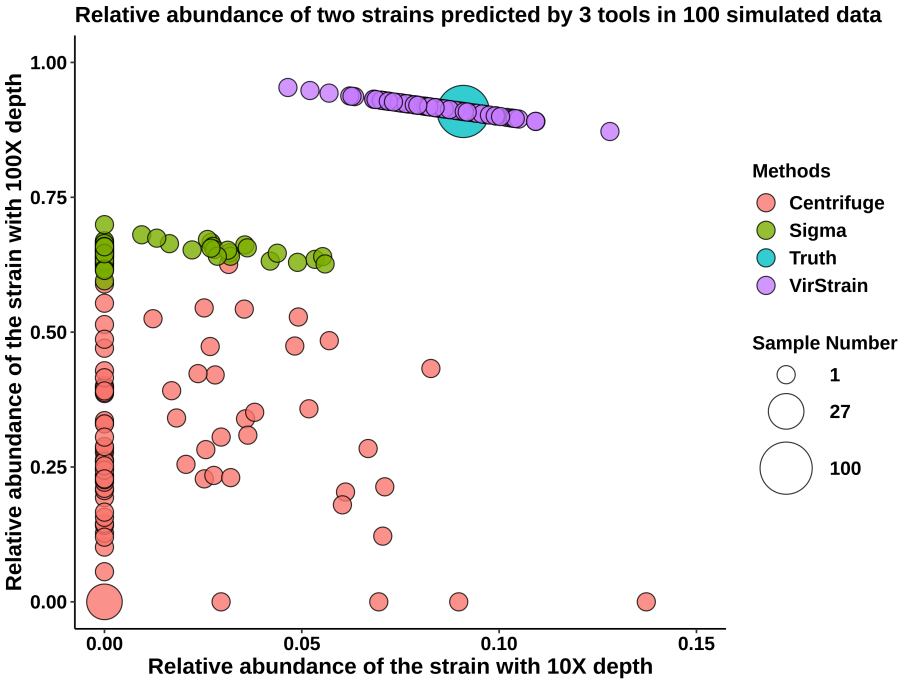

Figure 9: Predicted relative abundances across 100 simulated SARS-CoV-2 two-strain datasets for VirStrain, Sigma, and Centrifuge. The size of each point (circle) represents the number of simulated data sets. "Truth" refers to the ground truth of the relative abundance of the 2 strains in each data set, which is calculated by normalizing the sequencing depth (100X and 10X).

4.7 VirStrain detects the closest relative for novel strains

When a strain is not present in the reference database, VirStrain will output its closest relative in the database. Here, we define the closest relative as the strain in the database that is most similar to the query strain identified by MegaBLAST [40]. In order to test VirStrain on returning the closest relative for novel strains, we created multiple simulated read sets from mutant strain genomes.

In order to focus on testing the ability of different tools on detecting the

closest relative, we need to reconstruct our reference genome set by choosing only the sequences that can be correctly identified by all tools. Thus, we used 53 SARS-CoV-2 genome sequences that can be identified correctly by all tools in the single-strain experiment. Then, we used simuG program [39] to simulate random point mutations to each of these genomes. According to Dorp et al. [34], the average number of mutations in the SARS-CoV-2 strains is 9.6. Thus, we simulated mutant genomes with 5, 7, 9, 11, 13 random point mutations from the raw genome sequences and marked these newly obtained genomes as M5, M7, M9, M11, M13. In total, we have 265 (53*5) mutant genomes and 53 raw (i.e. reference) genomes. Then, we simulated short reads from these mutant and raw genomes using ART with depth of 100X, average insert size of 600 bp, and standard deviation of 150 bp. Thus we have 318 (265 mutant and 53 raw) datasets as inputs. Similar to other experiments, we used accuracy as the evaluation metric. For each dataset, as it only contains reads simulated from one strain, we thus only keep the top 1 output by different tools.

Figure 10 shows that VirStrain and Sigma are able to find the correct closest relatives in all data sets, while the other 3 tools failed to output the correct strains in some cases. This is consistent with the experimental results of Sigma [1], which tested this function on multiple datasets. This experiment demonstrated that the performance of our tool is as good as the mapping-based tool Sigma in identifying the closest relative. It is noteworthy that Figure 10 looks better than Figure 7 for several tested tools because this experiment only used 53 strains that are correctly identified by all tools in the single-strain experiment.

To further test the robustness of VirStrain, we applied VirStrain to detect the closest relative in a larger simulated dataset. Sigma was not included in this experiment because of its long running time. We randomly selected 100 strains from all the reference genomes. Then we generated 500 mutant strains from these 100 strains with 5, 7, 9, 11, 13 random point mutations. With ART, we simulated 600 datasets (for 500 mutant strains and 100 reference strains) and used them as input to VirStrain. VirStrain can still identify correct closest relatives in all the 600 simulated datasets (Supplementary Figure S7). Based on these results, we find that VirStrain is able to return the closest relatives for novel strains.

4.8 VirStrain detects SARS-CoV-2 strains from real sequencing data

To evaluate the performance of VirStrain for SARS-CoV-2 identification, we conducted experiments on a total of 11 real-world data sets. Out of 11 samples, there is only one sample with its reference genome collected in VirStrain database. These samples are sequenced with different platforms including Illumina, BGI-Seq and Ion Torrent, which makes it a good opportunity to test the robustness of our tool. For samples with complete reference genomes, we used blastn to find the most similar strain of their reference genomes in the VriStrain database and used it as the ground truth. The results are shown in Table 3.

Table 3 showed that the derived and known geographical information is

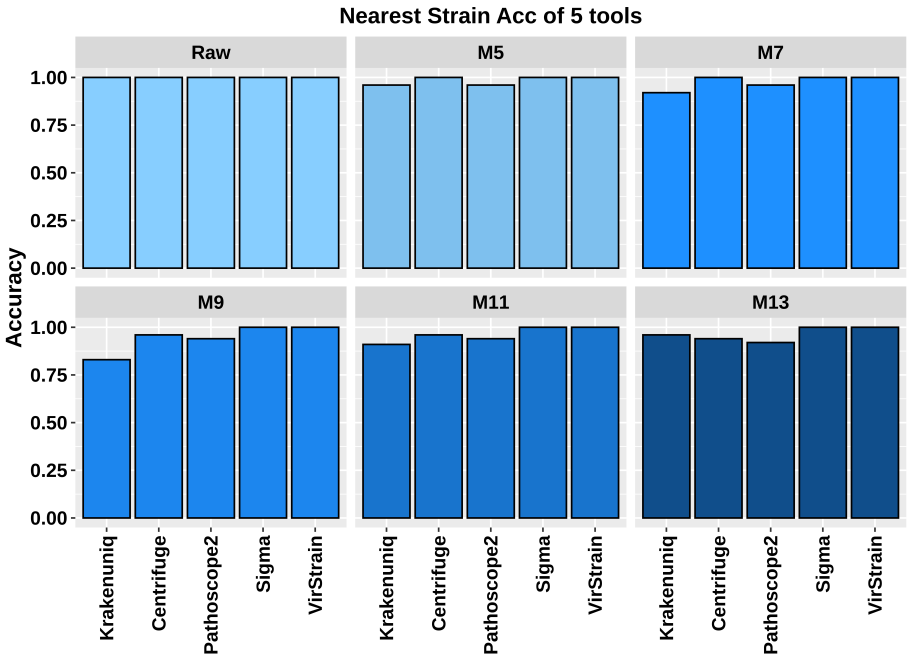

Figure 10: The accuracy comparison of 5 tools on detecting the closest relative in 318 simulated datasets. The 53 strains used in this experiment can be correctly identified by all tools in Figure 7. "Raw" means the dataset from the reference genomes and M5, M7, M9, M11, M13 represent datasets simulated from mutant strains. There are a total of 53 data sets for each group and each dataset contains one strain.

generally consistent for all datasets. For cases where the complete genomes are available, the strains returned by VirStrain are the same as the output of MegaBLAST.

The first sample SAMN13922059 is actually from a patient in Wuhan, China, whose sample was used to generate the first reference genome of SARS-CoV-2[37]. In VirStrain this first reference genome is actually located in a cluster with other 47 SARS-CoV-2 strains, which all belong to clade 19A defined by nextstrain. In this cluster, there are two main geographical locations: Wuhan and Hangzhou, China. As Hangzhou's cases are slightly more than Wuhan, we used Hangzhou in column "Region of clusters". This is one current limitations of VirStrain. These 48 strains cannot be divided into single-strain clusters.

There are 2 very interesting samples in Table 3: SAMN14560168 and SAMN14643484 (bold font). SAMN14560168 is from the first COVID-19 patient of Cambodia, who had been to China before being admitted to the hospital. The identification result of VirStrain shows its closest relative is MT470175.1, which is from

BioSample accession number	Sequencing platform	Blast result	VirStrain result	Data size	Running time	Region of sample	Region of clusters	Genome in the DB
SAMN13922059	Illumina MiniSeq	MT470175.1	MT470175.1	19.5 GB	215s	Wuhan, China	Hangzhou, China	Y
SAMN15941290	Ion Torrent S5	MT576531.1	MT576531.1	625 MB	19s	Gujarat, INDIA	Gujarat, INDIA	N
SAMEA7098097	Illumina MiSeq	MT520283.1	MT520283.1	1.2 GB	20s	Stockholm, Sweden	Massachusetts, USA	N
SAMN15916668	NextSeq 500	MT745701.1	MT745701.1	309 MB	14s	Victoria, Australia	Victoria, Australia	N
SAMN14560168	Illumina iSeq 100	MT470175.1	MT470175.1	2.8 GB	47s	Cambodia	Hangzhou, China	N
SAMN14643484	Illumina MiSeq	MT159710.2	MT159710.2	1.1 GB	31s	Israel	DP cruise ship	N
SAMN15144727	BGISEQ-500	Unknown	MT568634.1	32 MB	13s	Guangzhou, China	Guangzhou, China	N
SAMN15637956	Illumina HiSeq 4000	Unknown	MT066175.1	6.8 GB	114s	China	Guangzhou, China	N
SAMN16058334	NextSeq 500	Unknown	MT633030.1	8.2 GB	73s	Washington, USA	Washington, USA	N
SAMN16068354	NextSeq 500	Unknown	MT641532.1	612 MB	16s	Nevada, USA	Washington, USA	N
SAMN14668182	Ion Torrent S5	Unknown	MT704132.1	24 MB	13s	New York, USA	Maryland, USA	N

Table 3: The VirStrain identification result of 11 real sequencing datasets. "Unknown" in the column "Blast result" means that the complete genome of that dataset is not available. "Region of clusters" is the output of VirStrain based on the metadata associated with the reference strains in each cluster. For clusters containing more than one reference strain, we use the majority vote to get the geographical region information. "Genome in the DB" represents whether the complete genome of that dataset can be found in the reference database of VirStrain, yes (Y) or no (N). "DP cruise ship" refers to the Diamond Princess cruise ship.

China. Thus, this indicates that this Cambodia patient could be infected in China, which is consistent with this patient's travel history. Another interesting case SAMN14643484 is from Israel, and its closest relative identified by VirStrain is from the Diamond Princess cruise ship. According to the sample information in NCBI, this patent was a passenger of the Diamond Princess cruise ship and did get an infection there.

These results show that that VirStrain is able to identify SARS-CoV-2 strains from real sequencing data with or without assembled genomes. In addition, VirStrain also provides information that can be very useful for tracking virus spread and predicting clades.

4.9 VirStrain detects possible minor strains from Washington State datasets

For RNA viruses with fast replication rates, it is not rare to see multiple strains infecting the same host. For COVID-19, is it possible to see more than one strain (or haplotype) in the same host? Reconstructing the complete haplotype from short reads is an active research area itself and is beyond the scope of this work. But we can examine whether a different local haplotype segment exist in a sample. If we find one such local haplotype segment, we can add it to our reference sequence database and search for it in other samples using VirStrain. Being able to detect the same local haplotype in multiple samples provides strong evidence behind the presence of a new haplotype.

Towards this goal, we first mapped reads from sequenced SARS-CoV-2 samples at NCBI SRA against the first reference genome of SARS-CoV-2. We found some relatively high-frequency mutations in a Washington State sample (SAMN15678406) from USA (Supplementary Figure S1). Based on the results

of the reads mapping and a statistical model (see Supplementary Methods) that considers the base quality, sequencing errors, coverage, and the number of reads covering the same SNV events, we obtained a possible local region of a minor strain, which we named as k59_23. We used nextstrain [13] to annotate k59_23 and found that there is a mutation G25358C that will cause the amino acid mutation D1259H of the surface protein.

In order to validate this finding, we checked whether other samples also contain k59_23. Towards this goal, we added this strain to the VirStrain database and then used VirStrain to detect it in other SRA datasets from Washington State. The output shows that k59_23 was also detected in three other Washington State samples. The VirStrain detection results of these four data sets are summarized in Table 4. Both the statistical analysis and the VirStrain outputs show that a minor strain likely exists in these samples. This indicates the possible utility of VirStrain in detecting multi-infection cases.

BioSample accession number	VirStrain Result 1	Relative abundance 1	VirStrain Result 2	Relative abundance 2
SAMN15678406	MT641500.1	0.8523	k59_23	0.1477
SAMN15678405	MT375468.1	0.9042	k59_23	0.0958
SAMN15678404	MT632835.1	0.8797	k59_23	0.1203
SAMN15678403	MT628075.1	0.9006	k59_23	0.0994

Table 4: The identification results of VirStrain of four Washington State samples. VirStrain can identify two strains in these data sets. The numbers "1" and "2" represent these 2 strains. The major strains in these four samples returned by VirStrain are different but all have geographical locations from Washington State. The relative abundance of each strain is calculated by normalizing the depth of all identified strains.

4.10 VirStrain identifies 5 strains from HIV mock data

In this experiment, we applied VirStrain to a mock dataset (SRR961669) containing real sequencing data from five HIV strains. The authors mixed five HIV strains (JRCSF, 89.6, NL43, YU2, and HXB2) and conducted Illumina sequencing [10]. This dataset is frequently used for testing virus haplotype reconstruction algorithms. Using the reads as input, VirStrain can detect 5 strains from its reference database and predict their sequencing depth. Based on the predicted sequencing depth, we calculated the relative abundance by normalizing the depth of each identified strain.

To compare the predicted abundance with the ground truth, we applied the chi-square test and got the p-value 0.9998, which indicates that the distribution of the predicted abundance by VirStrain is not statistically different from the ground truth (Figure 11). This experiment demonstrates the ability of VirStrain in identifying multiple strains in one sample.

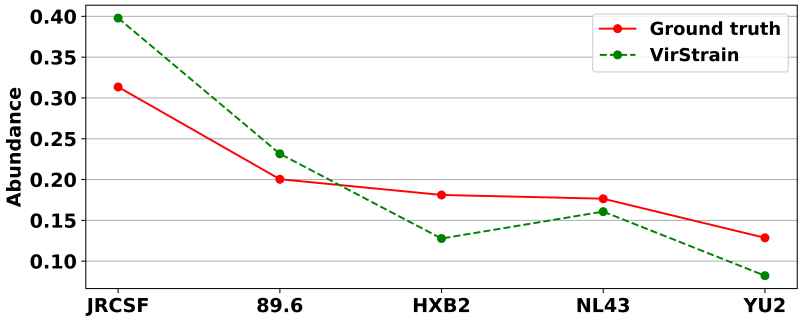

Figure 11: Abundance comparison of HIV mock data between the ground truth and VirStrain. The true average abundances sorted in descending order are: 31.35%, 20.04%, 18.11%, 17.65%, 12.86%

4.11 VirStrain predicts HIV subtype and H1N1 clade

The identification result generated by VirStrain can also be used to extract the subtype or clade information of identified strains. Thus, users can obtain the strain subtype or clade information from reads directly without assembly, which is useful for datasets that cannot form quality assemblies. In this experiment, we applied VirStrain to four real datasets and output the subtype or clade information of identified strains using a bottom-to-up method. To know the subtype or clade type of the input dataset, we can check the metadata of "the most possible strain" output by VirStrain. Then, we use the subtype or clade label of the identified strain as the final output.

BioSample accession number	Virus type	Subtype/Clade (Truth)	Subtype/Clade (VirStrain)	Region of sample	Major region of cluster
SAMN10225085	HIV	F1	F1	Brazil	Brazil
SAMN08101778	HIV	B	B	Brazil	Brazil
SAMN01095441	H1N1	1A.3.3.2	1A.3.3.2	Hong Kong	Hong Kong
SAMN02719578	H1N1	Other-Human	Other-Human	Unknown	Russia

Table 5: The VirStrain clade/Subtype prediction result of 4 real sequencing data. "Subtype": the subtype of HIV strain is obtained according to literature [2]. "Clade": the clade of H1N1 strain is estimated by the Swine H1 clade classification tool of Influenza Research Database (https://www.fludb.org).

Table 5 shows that the subtype or clade identified by VirStrain is the same as the truth. Besides, the region information is also consistent with the source region of these samples.

5 Discussion

In this work, we implemented a strain identification tool for short reads. We designed a greedy covering algorithm to divide reference genomes into multiple clusters so that the genomes in each cluster possess unique set of k-mers.

VirStrain shows higher accuracy than other tested tools across all benchmark datasets with different complexity. One of VirStrain's advantages is that it can detect the closest relative for novel strains. For example, VirStrain identified the most similar strain from datasets containing mutant genomes that are not in the database and demonstrated the same accuracy as using MegaBLAST [40] on complete genomes. Another advantage of VirStrain is to detect multi-strain infection cases. We demonstrated this by using VirStrain on both simulated and real sequencing data.

Currently we derive k-mers from aligned reference genomes. We noticed some alignment errors especially at sites with consecutive insertions or deletions. As a result, we tend to exclude columns with many indels, which may lead to clusters containing multiple genomes in the end. Ideally, we want to derive k-mer sets for reference genomes without relying on alignment programs, which is our future work.

We also plan to adapt this method for bacterial strain analysis, which is an important component in precise diagnosis and treatment for clinical data. Some bacteria already have a large number of sequenced strains, which can benefit from our fast k-mer set derivation method.

6 DATA AVAILABILITY

All data and codes used for this study are available online or upon request to the authors. The source code of VirStrain is freely available at https://github.com/liaoherui/VirStrain.

7 ACKNOWLEDGEMENTS

This work was supported by Hong Kong RGC GRF 9042828 and HKIDS (9360163).

References

[1] T. H. Ahn, J. Chai, and C. Pan. Sigma: strain-level inference of genomes from metagenomic analysis for biosurveillance. *Bioinformatics*, 31(2):170–177, 2015.

[2] B. M. Alves, J. D. Siqueira, I. M. Prellwitz, O. M. Botelho, V. P. Da Hora, S. Sanabani, P. Recordon-Pinson, H. Fleury, E. A. Soares, and M. A. Soares. Estimating HIV-1 Genetic Diversity in Brazil Through Next-Generation Sequencing. *Front Microbiol*, 10:749, 2019.

[3] C. Anyansi, A. Keo, B. J. Walker, T. J. Straub, A. L. Manson, A. M. Earl, and T. Abeel. QuantTB - a method to classify mixed Mycobacterium tuberculosis infections within whole genome sequencing data. *BMC Genomics*, 21(1):80, 2020.

[4] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler. GenBank. *Nucleic Acids Res*, 33:D34–38, 2005.

[5] F. P. Breitwieser, D. N. Baker, and S. L. Salzberg. KrakenUniq: confident and fast metagenomics classification using unique k-mer counts. *Genome Biol.*, 19(1):198, 2018.

[6] Jiao Chen, Jiating Huang, and Yanni Sun. TAR-VIR: a pipeline for TARgeted VIRal strain reconstruction from metagenomic data. *BMC Bioinformatics*, 20:305, 2019.

[7] Shifu Chen, Changshou He, Yingqiang Li, Zhicheng Li, and III Melançon, Charles E. A computational toolset for rapid identification of SARS-CoV-2, other viruses and microorganisms from sequencing data. *Briefings in Bioinformatics*, 2020.

[8] A. T. Dilthey, C. Jain, S. Koren, and A. M. Phillippy. Strain-level metagenomic assignment and compositional estimation for long reads with MetaMaps. *Nat Commun*, 10(1):3066, 2019.

[9] G. Dudas, L. M. Carvalho, T. Bedford, A. J. Tatem, G. Baele, N. R. Faria, D. J. Park, J. T. Ladner, A. Arias, D. Asogun, et al. Virus genomes reveal factors that spread and sustained the Ebola epidemic. *Nature*, 544(7650):309–315, 2017.

[10] F. D. Giallonardo, A. Töpfer, M. Rey, S. Prabhakaran, Y. Duport, C. Leemann, S. Schmutz, N. K. Campbell, B. Joos, M. R. Lecca, et al. Full-length haplotype reconstruction to infer the structure of heterogeneous virus populations. *Nucleic acids research*, 42(14):e115–e115, 2014.

[11] N. D. Grubaugh, J. T. Ladner, M. U. G. Kraemer, G. Dudas, A. L. Tan, K. Gangavarapu, M. R. Wiley, S. White, J. Thézé, D. M. Magnani, et al. Genomic epidemiology reveals multiple introductions of Zika virus into the United States. *Nature*, 546(7658):401–405, 2017.

[12] D. F. Gudbjartsson, A. Helgason, H. Jonsson, O. T. Magnusson, P. Melsted, G. L. Norddahl, J. Saemundsdottir, A. Sigurdsson, P. Sulem, A. B. Agustsdottir, et al. Spread of SARS-CoV-2 in the Icelandic Population. *N. Engl. J. Med.*, 382(24):2302–2315, 2020.

[13] J. Hadfield, C. Megill, S. M. Bell, J. Huddleston, B. Potter, C. Callender, P. Sagulenko, T. Bedford, and R. A. Neher. Nextstrain: real-time tracking of pathogen evolution. *Bioinformatics*, 34(23):4121–4123, 2018.

[14] N. Harel, M. Meir, U. Gophna, and A. Stern. Direct sequencing of RNA with MinION Nanopore: detecting mutations based on associations. *Nucleic Acids Res.*, 47(22):e148, 2019.

[15] C. Hong, S. Manimaran, Y. Shen, J. F. Perez-Rogers, A. L. Byrd, E. Castro-Nallar, K. A. Crandall, and W. E. Johnson. PathoScope 2.0: a complete computational framework for strain identification in environmental or clinical sequencing samples. *Microbiome*, 2:33, 2014.

[16] W. Huang, L. Li, J. R. Myers, and G. T. Marth. ART: a next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, 2012.

[17] M. R. Islam, M. N. Hoque, M. S. Rahman, A. S. M. R. U. Alam, M. Akther, J. A. Puspo, S. Akter, M. Sultana, K. A. Crandall, and M. Anwar Hossain. Genome-wide analysis of SARS-CoV-2 virus strains circulating worldwide implicates heterogeneity. *Sci Rep*, 10(14004), 2020.

[18] K. Katoh, K. Misawa, K. Kuma, and T. Miyata. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res*, 30(14):3059–3066, 2002.

[19] D. Kim, L. Song, F. P. Breitwieser, and S. L. Salzberg. Centrifuge: rapid and sensitive classification of metagenomic sequences. *Genome Res*, 26(12):1721–1729, 2016.

[20] M. Kiso, K. Mitamura, Y. Sakai-Tagawa, K. Shiraishi, C. Kawakami, K. Kimura, F. G. Hayden, N. Sugaya, and Y. Kawaoka. Resistant influenza A viruses in children treated with oseltamivir: descriptive study. *Lancet*, 364(9436):759–765, 2004.

[21] J. H. Kuhn, Y. Bao, S. Bavari, S. Becker, S. Bradfute, J. R. Brister, A. A. Bukreyev, K. Chandran, R. A. Davey, O. Dolnik, J. M. Dye, et al. Virus nomenclature below the species level: a standardized nomenclature for natural variants of viruses assigned to the family Filoviridae. *Arch Virol*, 158(1):301–311, 2013.

[22] J. T. Ladner, N. D. Grubaugh, O. G. Pybus, and K. G. Andersen. Precision epidemiology for infectious disease control. *Nat. Med.*, 25(2):206–211, 2019.

[23] C. Luo, R. Knight, H. Siljander, M. Knip, R. J. Xavier, and D. Gevers. ConStrains identifies microbial strains in metagenomic datasets. *Nat Biotechnol*, 33(10):1045–1052, 2015.

[24] G. Marcais and C. Kingsford. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, 27(6):764–770, 2011.

[25] S. E. Mate, J. R. Kugelman, T. G. Nyenswah, J. T. Ladner, M. R. Wiley, T. Cordier-Lassalle, A. Christie, G. P. Schroth, S. M. Gross, G. J. Davies-Wayne, et al. Molecular Evidence of Sexual Transmission of Ebola Virus. *N. Engl. J. Med.*, 373(25):2448–2454, 2015.

[26] R. A. Neher and T. Bedford. nextflu: real-time tracking of seasonal influenza virus evolution in humans. *Bioinformatics*, 31(21):3546–3548, 2015.

[27] L. Perrin and A. Telenti. HIV treatment failure: testing for HIV resistance in clinical practice. *Science*, 280(5371):1871–1873, 1998.

[28] A. F. Poon, R. Gustafson, P. Daly, L. Zerr, S. E. Demlow, J. Wong, C. K. Woods, R. S. Hogg, M. Krajden, D. Moore, et al. Near real-time monitoring of HIV transmission hotspots from routine HIV genotyping: an implementation case study. *Lancet HIV*, 3(5):e231–238, 2016.

[29] C. Quince, T. O. Delmont, S. Raguideau, J. Alneberg, A. E. Darling, G. Collins, and A. M. Eren. DESMAN: a new tool for de novo extraction of strains from metagenomes. *Genome Biol*, 18(1):181, 2017.

[30] M. Scholz, D. V. Ward, E. Pasolli, T. Tolio, M. Zolfo, F. Asnicar, D. T. Truong, A. Tett, A. L. Morrow, and N. Segata. Strain-level microbial epidemiology and population genomics from shotgun metagenomics. *Nat Methods*, 13(5):435–438, 2016.

[31] Y. Shu and J. McCauley. GISAID: Global initiative on sharing all influenza data - from vision to reality. *Euro Surveill*, 22(13), 2017.

[32] J. P. Skittrall, C. K. Ingemarsdotter, J. R. Gog, and A. M. L. Lever. A scale-free analysis of the HIV-1 genome demonstrates multiple conserved regions of structural and functional importance. *PLoS Comput Biol*, 15(9):e1007345, 2019.

[33] D. T. Truong, A. Tett, E. Pasolli, C. Huttenhower, and N. Segata. Microbial strain-level population structure and genetic diversity from metagenomes. *Genome Res*, 27(4):626–638, 2017.

[34] L. van Dorp, M. Acman, D. Richard, L. P. Shaw, C. E. Ford, L. Ormond, C. J. Owen, J. Pang, C. C. S. Tan, F. A. T. Boshier, et al. Emergence of genomic diversity and recurrent mutations in SARS-CoV-2. *Infect Genet Evol*, 83:104351, 2020.

[35] D. E. Wood and S. L. Salzberg. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biol.*, 15(3):R46, 2014.

[36] Lu J. Wood, D.E. and B. Langmead. Improved metagenomic analysis with Kraken 2. *Genome Biol*, 20(257), 2019.

[37] F. Wu, S. Zhao, B. Yu, Y. M. Chen, W. Wang, Z. G. Song, Y. Hu, Z. W. Tao, J. H. Tian, Y. Y. Pei, et al. A new coronavirus associated with human respiratory disease in China. *Nature*, 579:265–269, 2020.

[38] Y. Yan, L. H. Nguyen, E. A. Franzosa, and C. Huttenhower. Strain-level epidemiology of microbial communities and the human microbiome. *Genome Med*, 12(1):71, 2020.

[39] J.X. Yue and Gianni Liti. simuG: a general-purpose genome simulator. *Bioinformatics*, 35(21):4442–4444, 2019.

[40] Z. Zhang, S. Schwartz, L. Wagner, and W. Miller. A greedy algorithm for aligning DNA sequences. *J. Comput. Biol.*, 7(1-2):203–214, 2000.