# ReadSniper: Rapid Retrieval of Relevant Reads

Software for Finding Relatives of Viruses in DNA Sequence Datasets

Anton Oresten Sollman

anton.sollman@outlook.com

Initially part of Karolinska Institute's

Summer Research School in Computational Biology and Bioinformatics of 2022

Supervisor: Ben Murrell

Continued at Gullmarsgymnasiet

Supervisors: Jens Bjelvenmark, Pia Engström

May 9th, 2023

## Abstract

Viruses can exhibit incredible diversity, and it is not uncommon to discover a virus in a sample that bears little to no resemblance to anything found with standard tools and databases. The largest collection of sequencing data, the Sequence Read Archive (SRA), represents a potential source of related viruses, but computational limitations do not allow us to directly query the entire SRA with any given virus genome. For RNA viruses, Serratus is a recent effort that has built a repository of all RNA-dependent RNA polymerase (RdRp)-sequences found in the SRA, allowing us to find datasets that contain the RdRp of a given reference virus genome. To leverage this revolutionary capability, we have developed ReadSniper, which we have found to be capable of finding sequences in datasets that match to regions outside of just the RdRp, enabling the assembly of complete genomes of a reference virus' relatives.

## Table of Contents

## 1 Introduction

The increasing efficiency of DNA sequencing methods is generating biological sequence data at faster than an exponential rate, and the current size of public databases is exceeding 20 quadrillion base pairs (Edgar, et al., 2022). Larger-scale analysis of genomes has only been made possible in recent decades, with advancements in sequencing and computational speeds. We are however reaching a point where computational speed alone is cannot match the flux of new data. New and efficient methods for different use cases are therefore crucial for handling the large sets of data being generated.

This paper will cover and discuss the methods and algorithms used for ReadSniper: a program that aims to efficiently scan DNA sequence datasets for approximate matches with a reference DNA sequence, such as a reverse-transcribed RNA virus genome. This could be useful in cases where a virus (or close relative of one) is expected to have been present in a sequenced sample. Finding related reads is not a trivial task, as sequenced samples may have contained a variety of other unrelated RNA as well. ReadSniper attempts to identify the sequences of a dataset that match to a reference sequence and ignores the sequences whose scores do not exceed a certain threshold, such that the related sequences can later be assembled through *de novo assembly*[1] and then compared with the original reference sequence.

ReadSniper cannot directly query the millions of datasets in the Sequence Read Archive (SRA). Instead, it can make use of Serratus, which has already indexed entire the SRA using three catalytic motifs of the RNA-dependent RNA polymerase (RdRp) gene (Edgar, et al., 2022). Along with ReadSniper's methods and algorithms, this paper will cover one way of leveraging the capabilities of Serratus' palmID by showing and discussing the results from a trial run of ReadSniper coupled with palmID on an unknown and unclassified 9366 base pair RNA virus.

### 1.1 Sequencing, datasets, and reads

DNA sequence datasets can be retrieved from the Sequence Read Archive (SRA) using the SRA Toolkit (SRA-Toolkit, 2022). These datasets are produced from sequencing runs, which analyze DNA fragments and determine the order of the four bases: A, C, G, and T, which gets reported as a series of strings called *reads* (DNA sequencer, 2022). The length of a read is the number of bases sequenced from a particular DNA fragment. Shorter reads are more cost-effective, and sufficient for most applications, although longer reads allow for more sequence overlap between sequenced DNA fragments (Illumina, n.d.). The SRA contains millions of these datasets, with each dataset coming from a sequencing run performed on a sample from an experiment.

Modern paired-end sequencing technology takes advantage of both ends of double-stranded DNA by reading DNA molecules from both ends, producing twice the amount of data per sequencing cycle (Illumina, n.d.). Double-stranded DNA consists of base pairs with complementary bases (Figure 1). Illumina sequencing reagents feature a certain number of sequencing cycles, which directly relates to the read length since one base is sequenced per cycle. A 300-cycle kit could be used for a 1 × 300 base pair single-read run or a 2 × 150 base pair paired-end run (Illumina, n.d.). This means that for paired reads, there will be sequence

[1] De novo assembly: reconstructing a longer sequence from short fragments, without the use of a reference sequence.

overlap in the middle region of that if the combined read length exceeds the length of the sequenced DNA fragment (Figure 2b).

```
A=T
T=A   5'-ATGCGCTAGCTCATTT-3'
G≡C   3'-TACGCGATCGAGTAAA-5'
C≡G
```

**Figure 1:** Double-stranded DNA. The two strands are antiparallel and complementary, which means that one strand will be the *reverse complement* of the other.

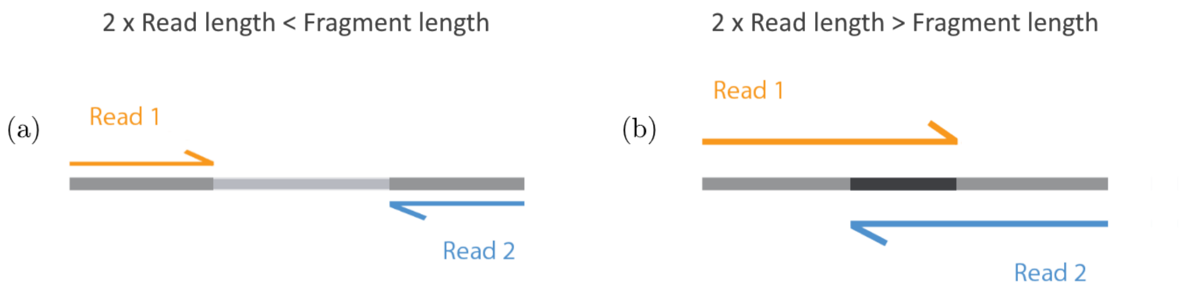
2 x Read length < Fragment length
2 x Read length > Fragment length
(a)
Read 1
Read 2
(b)
Read 1
Read 2

**Figure 2: a:** Paired-end sequencing will sequence both ends of a double-stranded DNA fragment, and the reads will not overlap if the total length is less than the length of the fragment. **b:** The reads may be long enough for a region in the middle region to get sequenced twice; the total length is thus greater than the length of the fragment.

In order for RNA to be studied using DNA-based techniques, it needs to be isolated and converted to *complementary* DNA (cDNA). These techniques and manipulations most often involve DNA rather than RNA as it is a more stable molecule (RNA-Seq, 2023). RNA is converted to DNA through *reverse transcription*, which creates double-stranded DNA from single-stranded RNA. The DNA must then be fragmented, as sequencers can only read a certain number of bases from the ends of a DNA molecule. After this, PCR can be performed for amplification of the DNA to make sequencing easier (Reverse transcriptase, 2022). Since the DNA is double-stranded, it can be read in both directions using paired-end sequencing. A read from a read pair may therefore match to a reference sequence in one direction, while the other read matches to that same reference sequence in the other (reverse complement) direction (Figure 3).

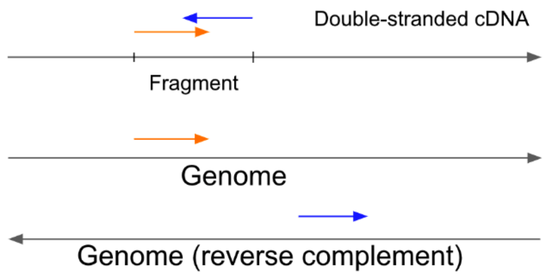

**Figure 3:** A fragment of a double-stranded cDNA molecule is sequenced into two reads. Since the two reads go in opposite directions, they would match to the reference genome in different directions.

## 1.2 Useful tools for identifying viruses

One of many non-trivial problems in bioinformatics is finding matching regions between sets of biological sequences. There are a number of tools that attempt to tackle this problem in different ways. BLAST is a widely used and highly capable tool with many applications, including identifying sequences and listing similar ones (BLAST (biotechnology), 2022). However, BLAST's methods alone are not enough to fully analyze the sheer amount of data and genetic diversity. If, for example, no close match of a query virus genome is found in a standard virus database, then it might be necessary to analyze public sequencing data to find a relative.

Serratus is a recent and ambitious effort that analyzes the enormous amount of data in the Sequence Read Archive (SRA). Serratus has built a repository (palmDB) of 883,502 RNA-dependent RNA polymerase (RdRp)-containing sequences, including RdRp from 131,957 novel RNA viruses (sequences which diverge from any previously known RdRp by over 10%) (Edgar, et al., 2022). RdRp is an enzyme that catalyzes the replication of RNA from an RNA template and is universal to RNA viruses of the *Orthornavirae* kingdom (RNA-dependent RNA polymerase, 2022), which makes it a useful marker for understanding their evolution. Serratus' palmID (palmID: Viral-RdRP Analysis, n.d.) can be used to find datasets containing RdRp-sequences similar to a given RNA virus' RdRp-sequence.

## 1.3 ReadSniper

It would require a tremendous amount of time to search through every dataset in the SRA for a single virus genome. ReadSniper can leverage Serratus' ability of reducing the number of datasets that need to be searched, by only scanning the relevant datasets for not just an RdRp-sequence, but a whole virus genome. ReadSniper can retrieve (or snipe) reads from a dataset that prove to have a significant correlation with a reference sequence. This shrinks the set of reads that a genome assembler has to piece together by several orders of magnitude. A de novo assembly of a genome similar to that of the reference is important because if it is possible to recreate a sequence similar to the reference genome using reads from just one dataset, then it has essentially been proven that the dataset came from a sample that contained RNA from a close relative of the reference virus. After a number of genomes have been assembled from different datasets, a phylogenetic tree can be constructed which makes it easier to assess the evolution of the genomes by considering the source of the sequenced samples.

ReadSniper is a package written in Julia: a fast dynamic programming language specifically designed to be fast and efficient for numerical and scientific computing. Julia has an extensive ecosystem of packages that are useful for biological computing (BioJulia, n.d.), including BioSequences and FASTX, which together form a streamlined set of functions for reading, writing, and analyzing FASTA and FASTQ-formatted datasets efficiently. Julia offers performance matching that of strongly typed compiled languages such as C and C++, while also being easy to use by providing interactive dynamic behavior. Julia is often a hundred times faster than interpreted languages such as Python, due to being designed ground up to take advantage of efficient and modern techniques (Bezanson, Karpinski, Shah, & Edelman, 2012). One quirk about Julia is that array indexing starts at 1, rather than 0. This changes some expressions to be slightly different when compared to how they would be written in other languages. For this reason, indices will also start at 1 in this paper. The ReadSniper Julia package is open-source, and available on GitHub (Periareion, 2022).

## 2 Method

ReadSniper uses a k-mer-based approach for finding matching regions between the reference sequence and the reads. k-mers are subsequences of length $k$ contained within a biological sequence (k-mer, 2022).[2] This means that there will be $L - k + 1$ k-mers in a sequence of length $L$ (Figure 4), some of which might be identical. In this context, k-mers represent oligonucleotides. Since there are four nucleotides in DNA, there are a total of $4^k$ possible unique k-mers (DNA subsequences of length $k$). k-mers are useful for getting past artifacts such as mutations and inaccurate base calls in reads. Instead of calculating the edit distance between two longer sequences (a task which can be computationally heavy), we can look for exact k-mer matches between them. The set of k-mers that two sequences share in common may be large enough to indicate a significant correlation between them. In addition, we can also look at the order in which these k-mers occur.

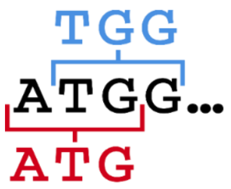

**Figure 4:** The first two 3-mers of some sequence that starts with "ATGG" are "ATG" and "TGG" (k-mer, 2022).

### 2.1 Finding relevant datasets

Serratus' palmID takes a query RNA virus' RdRp-sequence and returns an extensive report which among other things contains a list of datasets that have all been shown to contain an RdRp sequence similar to that of the RdRp found in the reference sequence. Serratus has indexed the RdRp sequences in each dataset beforehand, allowing it to quickly give us information regarding the correlation between the query RdRp and the RdRp found in datasets. In the list of datasets, Serratus also provides three metrics (*percent identity/pident*, *coverage*, and *expect value/evalue*) that are useful for assessing the relevance of each RdRp match. Relevant datasets are chosen based on the results from Serratus and fetched from the SRA using the open-source SRA Toolkit (SRA-Toolkit, 2022).

### 2.2 Reference sequence

ReadSniper uses a reference sequence (such as a virus genome) when searching through datasets. It creates a *k-mer dictionary* based on the reference sequence (: ). Since k-mers from a dataset need to be compared with k-mers from the reference sequence, it is crucial to make that comparison as fast as possible. Hash-based dictionaries provide a way to retrieve values using a key (of arbitrary datatype) with a *constant* look-up time with respect to the length of the reference sequence. In this case, the keys are k-mers represented as strings, and the values are indices represented as lists of numbers that point to where in the reference sequence the k-mer occurs (: ). Since dictionaries use hashes, the time it takes to find the indices of a k-mer in the reference sequence is constant, and *not* proportional to the number of k-mers in the reference, which would be the case for a naïve method that compares every k-mer of a read to every k-mer of the reference sequence.

[2] *mer*, like in *polymer*, "many parts", where $k$ is the number of nucleotides.

**Table 1:** The k-mer dictionary of the reference sequence[3] "ACGTACGT", with k = 4. The sequence has five overlapping subsequences of length k, four of which are unique, since "ACGT" occurs twice (at index 1 and 5). Every k-mer has a list of indices associated with them, pointing to where they occur in the reference sequence.

| k-mer | Index |
|---|---|
| ACGT | 1, 5 |
| CGTA | 2, |
| GTAC | 3 |
| TACG | 4 |

Since it is unknown from which direction a particular region of a DNA fragment was sequenced, the dictionary should also account for the reverse complement of each k-mer. This is done by appending negative indices to the list associated with the reverse complement of each k-mer (: ).

**Table 2:** The k-mer dictionary of the reference sequence "ATCGATCA", with k = 3. This version includes negative indices, which account for the reverse complement of the sequence. "TGA" does not occur in the original sequence, but it does occur in the reverse complement, "TGATCGAT", since it is the reverse complement of the k-mer at index 6 in the original sequence, "TCA".

| k-mer | Index |
|---|---|
| ATC | 1, -4, 5 |
| TCG | 2, -3 |
| CGA | -2, 3 |
| GAT | -1, 4, -5 |
| TCA | 6 |
| TGA | -6 |

The number of unique k-mers in a reference sequence may vary a lot depending on how repetitive it is. A non-repetitive reference sequence, such as one that has been randomly generated, may for example not have any significant repeats, whereas an authentic virus genome may have more and longer repeats as the sequence encodes real amino acids that relate to the structure of actual proteins.

## 2.3 Finding significant reads

### 2.3.1 k-mer matching and read scoring

Reads are assigned scores based on the number of k-mer matches within a certain window of the reference sequence, and the order they occur in. A read of length $L$ that matches with a region of the reference can be detected when there is a certain density of k-mer matches. Outside of this region, the density is expected to be much lower. It is likely that there will be some matches outside of this window that occur out of random chance, considering the limited number of possible k-mers (Figure 5). Moreover, there may be more than just $L-k+1$ matches with the reference, as a k-mer in the k-mer dictionary may point to multiple indices

[3] Reference sequences can be thousands of bases in length and will therefore have thousands of k-mers.

in the reference (: ). For any significant read, however, most matches are expected to be concentrated in an $L$-sized window of the reference. In the case where a reference sequence has repeats of a subsequence, a read may match to multiple regions. To find the most significant window of reference matches, ReadSniper searches for the *longest increasing subsequence* whose range of values is not greater than the read length (such as the lines in Figure 5). The random matches outside of the region are ignored and thus do not count towards the read's score. The start and end indices of these regions and the number of matches contained within them are later useful for visualizing read matches on a larger scale.

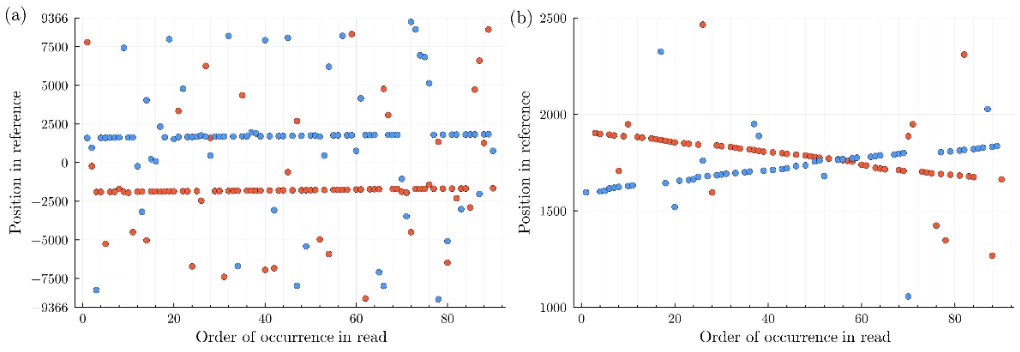

**Figure 5:** k-mer matches mapped onto a scatter plot. The y-coordinates represent where in the reference the k-mers from a read have matched to. The x-coordinates are the k-mers' order of occurrence in a read. The blue points are the k-mer matches from one read in a read pair, and the orange points are the k-mer matches from the other read in that same read pair. **a:** Negative y-coordinates represent matches to the reverse complement. **b:** Plotting the points with the absolute values of the y-coordinates show how the reads from the read pair overlap (as in Figure 2b).

When a read sequence pair matches with the reference sequence, straight lines are expected to appear in the k-mer match scatter plot, as in Figure 5, where the lines strongly suggest that two reads have matched to a region of the reference sequence. One read (orange) matched to the reverse complement of the reference sequence because it was read from the reverse complement direction of the reference sequence, whereas the other read (blue) was read from the opposite direction and thus matched to the reference sequence in the normal direction. Figure 5 shows how two reads may overlap if the combined read length is longer than the length of the DNA fragment that was sequenced, as shown in Figure 2b.

The time it takes to process a read sequence is dependent on the amount of k-mers sampled from the sequence. For this reason, ReadSniper has a *step* parameter, which increases the step between read k-mers. The step value is inversely proportional to the number of k-mers that get compared with the reference sequence. Increasing the step value leads to an increase in speed, at the cost of a smaller sample size. Furthermore, a higher step value has diminishing returns in terms of speed, as there are still parts of the read processing algorithm that are not influenced by this change.

### 2.3.2 Expected score distribution

ReadSniper estimates the average distribution of read scores that a non-correlated dataset would have. Based on such a distribution, a score threshold can be calculated for identifying outliers. Reads with scores that surpass this threshold are statistically significant and are saved to a list. This way, the results from millions of non-significant reads get discarded, and do not clutter the limited computer memory.

The probability of two random k-mers matching is needed to calculate the expected distribution of read scores. It is equivalent to the probability of $k$ random pairs of nucleotides all matching:

$$\mathrm{P(k\text{-}mer\ match)} = \mathrm{P(nucleotide\ match)}^{k}$$

ReadSniper considers the GC-contents of both the reference sequence and the dataset when calculating the probability of two nucleotides matching.

$$\mathrm{P(nucleotide\ match)} = \mathrm{P(both\ A)} + \mathrm{P(both\ C)} + \mathrm{P(both\ G)} + \mathrm{P(both\ T)}$$

$$\begin{aligned}
\mathrm{P(both\ A)} &= \frac{1-\mathrm{GC}_1}{2}\cdot\frac{1-\mathrm{GC}_2}{2}\\
\mathrm{P(both\ C)} &= \frac{\mathrm{GC}_1}{2}\cdot\frac{\mathrm{GC}_2}{2}\\
\mathrm{P(both\ G)} &= \frac{\mathrm{GC}_1}{2}\cdot\frac{\mathrm{GC}_2}{2}\\
\mathrm{P(both\ T)} &= \frac{1-\mathrm{GC}_1}{2}\cdot\frac{1-\mathrm{GC}_2}{2}
\end{aligned}$$

where $\mathrm{GC}_1$ is the GC-content of the reference and $\mathrm{GC}_2$ is the average GC-content of the dataset. The complement probability of a nucleotide being G or C (the GC-content) is the probability of it being either A or T. The probabilities are divided by two because G and C each have a 50% chance of occurring if the nucleotide is G or C. Likewise, A and T each also have a 50% chance of occurring if the nucleotide is A or T.

A list of probabilities for each score is calculated using the binomial formula:

$$\mathrm{P(score{=}m)} = \binom{n}{m} p^{m} q^{n-m}$$

where $n$ is the amount of k-mers in a read-length-sized range of the reference, $m$ is the longest increasing subsequence in a read-length-sized window of the k-mer index list, $p$ is the probability of a k-mer matching in a read-length-size window of the reference, and $q$ is the complement of $p$.

Minor adjustments are made based on empirical testing, as ReadSniper uses the *longest increasing subsequence* for scoring, which complicates the probability calculations considerably.

The formula can be shown to be accurate when plotted along with a distribution obtained empirically (Figure 6). That is, the distribution of scores that is found when running ReadSniper with a randomized dummy sequence. The read score threshold is arbitrarily assigned to be the score at which the expected amount of randomly matched reads is below some value. If that value is 1, the threshold will be the $(y = 10^{0})$-intersect of the expected score distribution in Figure 6. Every read with a read score less than that threshold is filtered out, and every read with a read score greater than or equal to that threshold is saved. The threshold is adjusted to account for the step size, since a higher step leads to a smaller sample size and consequently a lower score (inversely proportional to the step).

For datasets with reads related to the reference sequence, the distribution is expected to look different (Figure 6a), as some reads will get a higher score. The majority of non-correlated reads in Figure 6a are expected to have scores below the threshold (hence the spike), as there will most likely be other unrelated sequences in the sequenced samples. The threshold acts as a filter for filtering out the reads in the lower score range.

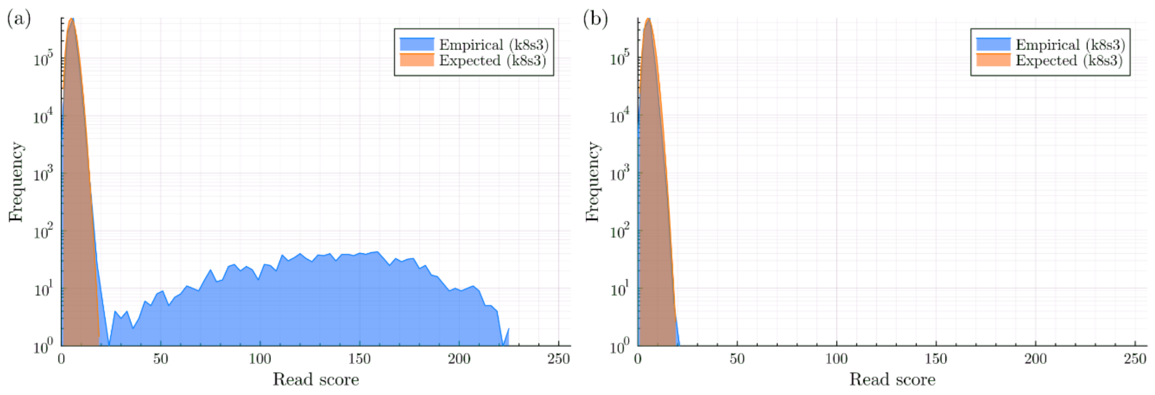

**Figure 6:** The empirical distributions of read scores (blue), and the expected distributions of read scores (orange). "k8s3" means that the run used a k-value of 8, and a step size of 3, both of which affect the shape of the distribution. **a:** The score distribution of reads from running ReadSniper with a virus genome that was expected to be found in a dataset. **b:** The score distribution of reads from running ReadSniper with a non-correlated *randomized* reference "genome" with the same GC-content. Note that the y-axis is logarithmic, and that the spike contains millions of reads. The x-axis range ends at 250, because it is roughly the theoretical maximum score for a read of length 250.

## 2.4 Coverage

After processing a dataset, the ranges from the matched reads get overlapped to produce a graph that visualizes sequence coverage. Additionally, a color gradient is used to show the scores of each read (Figure 7). When a reference genome is present in a dataset, the coverage plot is darker and covers most of the reference. Full genomic coverage is not guaranteed, however, since different genes may have different levels of divergence.

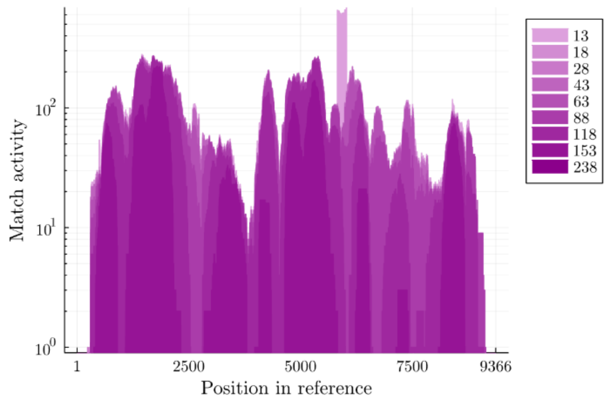

**Figure 7:** A logarithmic sequence coverage plot, with matched reads having been overlapped and mapped to their respective ranges in the reference sequence. Reads with higher score are represented with darker colors. The legend in the top right corner shows the score threshold for each color.

## 3 Results

The viability of ReadSniper has been assessed through testing on a particular RNA virus genome (9366 base pairs in length). *Nucleotide BLAST*, when optimized for "somewhat similar sequences (blastn)", found only a few similar genomes in the "Nucleotide collection (nr/nt)" database. These genomes matched to the reference genome with around 70% nucleotide identity, most of which were classified under different families in the *Picornavirales* order. Specifically searching for the RdRp returned a 65.25% amino acid identity match with a *Picornaviridae* RdRp.

Serratus found one RdRp match on species level in the palmDB database, two on genus level, and more than a hundred on family level (Figure 8). For each of these RdRp sequences, Serratus revealed multiple SRA datasets that might contain them.

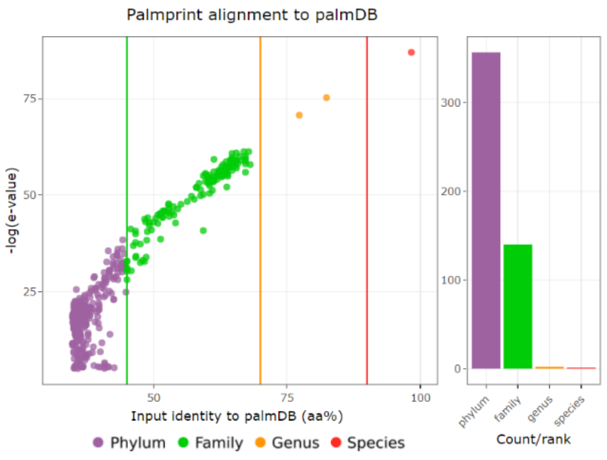

**Figure 8:** Serratus' palmID taxonomic results. One RdRp match is on species level (>90% amino acid identity), and two are on genus level (>70% amino acid identity) (palmID: Viral-RdRP Analysis, n.d.)

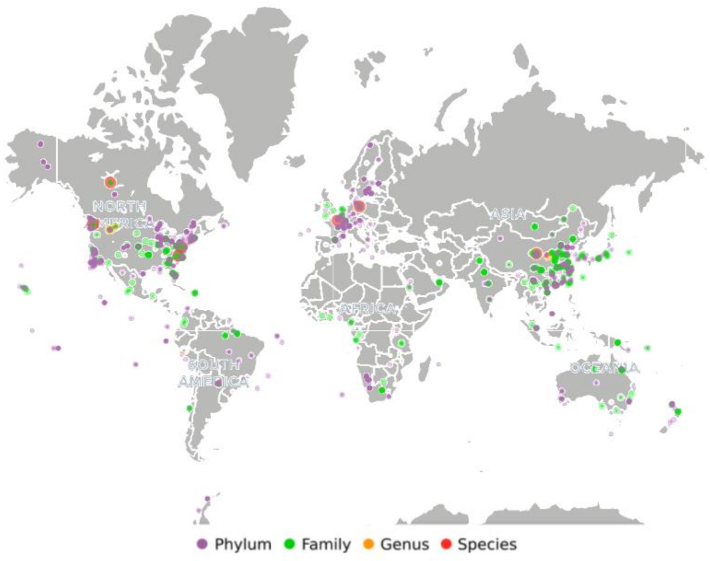

**Figure 9:** The geospatial distribution of matched RdRp found by Serratus' palmID. (palmID: Viral-RdRP Analysis, n.d.)

The datasets containing the species level RdRp matches were downloaded using the SRA Toolkit (SRA-Toolkit, 2022) and mined using ReadSniper to find the reads of each dataset that match with the reference genome. Only three out of ten datasets yielded enough reads for assembly. The reads from those three datasets were assembled into longer genomes using SPAdes, an open-source genome assembler (Center for Algorithmic Biotechnology, 2022). A phylogram was constructed based on the relative distances between the assembled genomes (Figure 10).

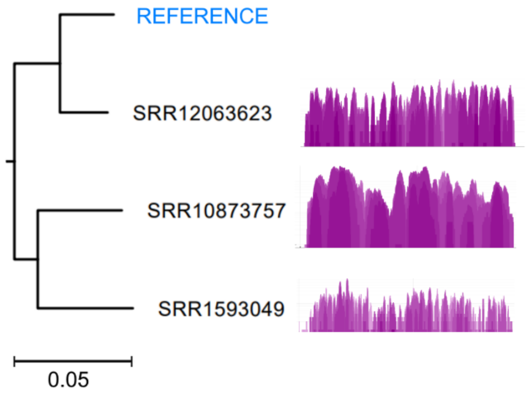

**Figure 10:** A phylogenetic tree of assembled genomes, and their respective coverages compared to the reference. The divergence between two genomes is given by the distance travelled on the x-axis, and the scale at the bottom which shows the divergence per unit length.

## 4 Discussion

### 4.1 Trial run on an unknown virus

The RNA virus genome used for this trial run of ReadSniper is unknown and confidential; details about it cannot be shared publicly. Serratus' palmID revealed that the unknown virus may have had relatives present in a variety of samples from around the world (Figure 9). ReadSniper found genome-wide matches in three out of the ten datasets containing the species level RdRp match (Figure 10). A dataset is considered to be a match to the reference if the read scores are sufficiently high and cover most of the reference. The dataset "SRR10873757" originates from a study on the virome of wild and breeding birds in China (Shan, et al., 2022), and the sample that was sequenced was a cloacal swab of a Red-flanked Bluetail (SRX7543514, 2020). "SRR1593049" was sequenced from a sample coming from a study that collected ballast water samples from bulk carriers in the Great Lakes of North America (SRX717357, 2014). "SRR12063623" was sequenced from a sample of pond water in Germany (SRX8591378, 2020). Each of the three assembled genomes had a nucleotide divergence of no greater than 10% from the reference virus genome (Figure 10).

The BLAST, Serratus, and ReadSniper results suggest that the virus belongs to an unclassified family in the order *Picornavirales* and has close relatives that have been found in different metagenome samples spanning multiple continents. The virus is novel, as it has no close matches in standard databases. The other seven datasets containing species level RdRp matches did not contain enough similar sequences to be assembled.

## 4.2 Strengths and limitations

When utilizing multiple CPU threads, ReadSniper can run at a speed on the order of a few seconds per gigabyte of FASTA-formatted files, but this speed will vary depending on the k-value used for k-mers. A k-value of 8 or 9 has shown to be most efficient[4], with the majority of the runtime then being file reading/writing and code compilation. The runtime is however proportional to the number of k-mer matches per read squared, which is not optimal, since the number of k-mers increases exponentially as the k-value is lowered. This time complexity is caused by one of the main algorithms used for processing the read matches, the *longest increasing subsequence*, which has a time complexity of $O(n^2)$. All this means that ReadSniper will be significantly slower when a lower k-value is used. Lower k-values are useful, however, because of the increased tolerance of mutations.

We have found that ReadSniper might only be suitable for datasets with higher-pident RdRp matches. ReadSniper only proved to be useful on datasets with a 98.3 amino acid RdRp pident, as it was unable to find any significant matches between the reference sequence and the next highest pident RdRp, with an amino acid pident of 82.4 (genus level), which sets a lower bound for the required RdRp pident. This could be due to the difference between amino acid pident and the nucleotide pident. Two similar amino acid sequences can be translated from two dissimilar nucleotide sequences. Similarly, the k-mer match rate is not necessarily representative of the nucleotide identity, as the same number of k-mer matches could either be densely concentrated in a small region (which could mean that the read has a lower nucleotide pident overall), or sparsely spread out across a whole read (which could indicate a higher nucleotide pident overall, with few, yet consistent nucleotide mutations). Moreover, regions outside of the RdRp gene could have an even lower pident, considering RdRp is highly conserved.

ReadSniper, at its core, is a lite Julia package capable of performing a quick linear search of a dataset to find reads that are similar to subsequences in a reference sequence. It is not optimized for lower pident matches. In these cases, a more advanced local BLAST search on the datasets may find more reads. Moreover, there are other tools such as NextGenMap, which, in addition to finding reads, also maps them more thoroughly onto a reference genome. NextGenMap uses k-mer hash-tables in a similar way to ReadSniper (Sedlazeck, Rescheneder, & von Haeseler, 2013). Comparisons with tools such as BLAST and NextGenMap need to be made in order to fully assess the viability of ReadSniper. Even if ReadSniper turns out to be slower or less capable, however, it still is a simple and perhaps practical Julia package which further extends the Julia package ecosystem.

## 4.3 Improvements

ReadSniper could be seamlessly made compatible with other Julia packages to create a more streamlined workflow. It might for example be practical to have the option to assemble the reads afterwards using a Julia de novo genome assembly package.

The current main scoring algorithm that uses the *longest increasing subsequence* is flawed because it finds the longest increasing subsequence *before* it checks that the range of values is less than the read length. As a result, it does not work as well when there are a lot of random k-mer matches, which would be the case when a low k-value is used. Any random increasing

[4] Odd values of k are however ideal since *odd*-mers cannot be their own reverse complements.

subsequence might be longer than the length of the true read match and might cover a window of the reference greater than the read length, which leads to scores being lower than they should. Implementing this new algorithm efficiently would be challenging, and it might even end up being considerably slower. There are other options for scoring, such as sorting the indices and finding the window with the most elements or calculating some overall density of matches across the reference. These approaches fail to consider both the order of the k-mer matches in the reference and the order they occur in the read, which leads to a higher rate of false matches. They might however provide a quicker way of seeding more thorough analysis of reads. An improved method would involve multiple steps per read, and process reads with increasingly finer levels of precision, as opposed to processing all of them fully in the same way.

Another way of visualizing the distances in Figure 10 could be to plot the genomes as a set of 2-dimensional points using multi-dimensional scaling, where the distances between the points may roughly correspond to the Levenshtein distances, for instance.

## 5 Acknowledgements

The ReadSniper project was initially worked on by me throughout the summer of 2022, as part of Karolinska Institute's first Summer Research School in Computational Biology and Bioinformatics. I would like to express sincere gratitude to my supervisor, Ben Murrell, who has offered invaluable support and continued guidance throughout the school year. The project would not have been possible without the organizers of the summer school: Ben Murrell, Gerry McInerney, and Claudia Kutter.

The project was continued at Gullmarsgymnasiet, Lysekil, as a senior year high school project throughout the Swedish school year of 2022-2023, with helpful feedback from Jens Bjelvenmark and Pia Engström.

## 6 References

Bezanson, J., Karpinski, S., Shah, V. B., & Edelman, A. (2012, September 24). *Julia: A Fast Dynamic Language for Technical Computing.* Retrieved from arXiv: https://arxiv.org/abs/1209.5145

*BioJulia.* (n.d.). Retrieved December 8, 2022, from BioJulia: https://biojulia.net/

*BLAST (biotechnology).* (2022, September 6). Retrieved November 10, 2022, from Wikipedia: https://en.wikipedia.org/w/index.php?title=BLAST_(biotechnology)&oldid=1108832330

Center for Algorithmic Biotechnology. (2022, July 16). *SPAdes.* Retrieved February 2, 2022, from Center for Algorithmic Biotechnology: http://cab.spbu.ru/software/spades/

*Complementarity (molecular biology).* (2022, November 24). Retrieved December 20, 2022, from Wikipedia: https://en.wikipedia.org/w/index.php?title=Complementarity_(molecular_biology)&oldid=1123588885

*DNA sequencer.* (2022, November 24). Retrieved November 24, 2022, from Wikipedia: https://en.wikipedia.org/w/index.php?title=DNA_sequencer&oldid=1123481270

Edgar, R. C., Taylor, J., Lin, V., Altman, T., Barbera, P., Meleshko, D., . . . Babaian, A. (2022, January 26). Petabase-scale sequence alignment catalyses viral discovery. *Nature, 602*, 142-147. Retrieved from Nature: https://www.nature.com/articles/s41586-021-04332-2

Illumina. (n.d.). *Read length recommendations.* Retrieved November 28, 2022, from Illumina: https://emea.illumina.com/science/technology/next-generation-sequencing/plan-experiments/read-length.html

*k-mer.* (2022, November 15). Retrieved January 19, 2023, from Wikipedia: https://en.wikipedia.org/w/index.php?title=K-mer&oldid=1122076985

*palmID: Viral-RdRp Analysis.* (n.d.). Retrieved December 12, 2022, from Serratus: https://serratus.io/palmid

Periareion. (2022). ReadSniper.jl. Retrieved from GitHub: https://github.com/Periareion/ReadSniper.jl

*Reverse transcriptase.* (2022, November 20). Retrieved December 14, 2022, from Wikipedia: https://en.wikipedia.org/w/index.php?title=Reverse_transcriptase&oldid=1122961256

*RNA-dependent RNA polymerase.* (2022, October 24). Retrieved November 17, 2022, from Wikipedia: https://en.wikipedia.org/w/index.php?title=RNA-dependent_RNA_polymerase&oldid=1118043833

*RNA-Seq.* (2023, February 6). Retrieved February 15, 2023, from Wikipedia: https://en.wikipedia.org/w/index.php?title=RNA-Seq&oldid=1137762560

Sedlazeck, F. J., Rescheneder, P., & von Haeseler, A. (2013, November 1). NextGenMap: fast and accurate read mapping in highly polymorphic genomes. *Bioinformatics, 29*(21), 2790-2791. Retrieved from Oxford Academic: https://academic.oup.com/bioinformatics/article/29/21/2790/195626

*Sequence Read Archive.* (n.d.). Retrieved December 8, 2022, from National Center for Biotechnology Information: https://www.ncbi.nlm.nih.gov/sra

Shan, T., Shixing, Y., Wang, H., Wang, H., Zhang, J., Gong, G., . . . Wang, X. (2022, April 12). Virome in the cloaca of wild and breeding birds revealed a diversity of significant viruses. *Microbiome, 10.* Retrieved from https://microbiomejournal.biomedcentral.com/articles/10.1186/s40168-022-01246-7

*SRA-Toolkit.* (2022, October 3). Retrieved November 28, 2022, from BIOWULF: https://hpc.nih.gov/apps/sratoolkit.html

*SRX717357.* (2014, October 9). Retrieved from National Library of Medicine: https://www.ncbi.nlm.nih.gov/sra/SRX717357

*SRX7543514.* (2020, January 12). Retrieved from National Library of Medicine: https://www.ncbi.nlm.nih.gov/sra/SRX7543514

*SRX8591378.* (2020, June 22). Retrieved from National Library of Medicine: https://www.ncbi.nlm.nih.gov/sra/SRX8591378