Article Paper

The utility of data transformation for alignment, *de novo* assembly and classification of short read virus sequences

Avraam Tapinos[1,*], Bede Constantinides[1,3], My VT Phan[2], Samaneh Kouchaki[1,4], Matthew Cotten[2], and David L Robertson[1,5]

[1] School of Biological Sciences, The University of Manchester, Manchester, UK
[2] Department of Viroscience, Erasmus Medical Centre, Rotterdam, the Netherlands
[3] Nuffield Department of Medicine, University of Oxford, Oxford, UK
[4] Department of Engineering Science, University of Oxford, Oxford, UK
[5] MRC-University of Glasgow Centre for Virus Research, Glasgow, UK

* Correspondence avraam.tapinos@manchester.ac.uk; Tel.: +44 (0) 161 701 7563

Received: date; Accepted: date; Published: date

Abstract: Advances in DNA sequencing technology are facilitating genomic analyses of unprecedented scope and scale, widening the gap between our abilities to generate and fully exploit biological sequence data. Comparable analytical challenges are encountered in other data-intensive fields involving sequential data, such as signal processing, in which dimensionality reduction (i.e., compression) methods are routinely used to lessen the computational burden of analyses. In this work we explore the application of dimensionality reduction methods to numerically represent high-throughput sequence data for three important biological applications of virus sequence data: reference-based mapping, short sequence classification and *de novo* assembly. Despite using highly compressed sequence transformations to accelerate the processes, our sequence processing approach yielded comparable accuracy to existing approaches, and are ideally suited for sequences originating from highly diverse virus populations. We demonstrate the application of our methodology to both synthetic and real viral pathogen sequence data. Our results show that the use of highly compressed sequence approximations can provide accurate results and that useful analytical performance can be retained and even enhanced through appropriate dimensionality reduction of sequence data.

Keywords: Alignment, assembly, taxonomic classification, time series, data transformation, DWT, DFT, PAA, data compression, compressive genomics

1. Introduction

Next generation sequencing (NGS) enables massively parallel determination of nucleotide order within genetic material, making it possible to rapidly sequence the genomes of individuals, populations and metagenomic samples [1-5]. However, the sequences generated by these instruments are almost always considerably shorter in length than the genomic regions studied. Genomic analyses often begin with the process of sequence assembly, where sequence fragments (reads) are reconstructed into the larger sequences from which they originated. Computational methods play a vital role in the assembly of short reads, and a variety of assemblers and related tools have been developed in tandem with emerging sequencing platforms [6]. All subsequent analyses and investigations depend upon the quality, accuracy and speed of this crucial sequence assembly process.

There are many computational methods to generate consensus sequences representing the genomes of coexisting species in a sample. Such approaches includes seed-and-extend alignment methods using suffix array derivatives such as the Burrows-Wheeler Transform (BWT) for aligning short reads informed by a known reference sequence [7,8], and graph-based methods employing Overlap Layout Consensus (OLC) [9,10] and *de* Bruijn graphs of *k*-mers [11-13] for reference-free *de novo* sequence assembly. However, for sequencing projects to characterise genetic variation within populations (deep sequencing), metagenomics and pathogen discovery, the effectiveness of the aforementioned approaches varies considerably [14].

Samples with mixed viral infections, especially those comprising divergent variants, present a number of analytical and computational problems. The use of a reference sequence, even the use of a data specific generated sequence, can lead to valuable read information being discarded during the alignment process [15]. On the other hand, while *de novo* approaches require little *a priori* knowledge of target sequence composition, the methods are computationally intensive and their performance scales poorly with datasets of increasing size [9]. Aggressive heuristics must be employed, to traverse graphs and deal with mismatches, to reduce the running time of *de novo* assemblers, which in turn can compromise assembly quality. Indexing structures such as the BWT and FM-index are widely used to reduce the burden of pairwise sequence comparison, for both reference base mapping and *de novo* assembly. However, they cannot process mismatches within reads, necessitating the use of aggressive and computationally expensive heuristics to establish relationships on divergent reads. The increasing NGS read length will affect the performance of these approaches [16].

A major challenge working with high-throughput sequencing data for metagenomics and within-host variation analysis is the substantially great diversity of biological data. Also the amount of sequences generated challenge many computational system for a feasible working solution in terms of time and the computational resources typically available in biological laboratories. For biologists working in outbreak responses or pathogen discovery, both the accuracy of the assembly results and the speed of sequence analyses (e.g. assembly, alignment and pathogen classification) are crucial for crisis response and management. The ability to run analyses in the field on portable computer systems without internet connectivity is also important. Here, we explore the utility of data transform methods (used in data intensive fields such as signal and image processing) to extract major features from viral NGS sequence data and use the features to analyse data in a lower dimensional space.

Similar analytical challenges involving high dimensional sequential data are encountered in other data-intensive fields such as signal and image processing, and time series analysis, where data transforms and approximation techniques are used for data dimensionality reduction. Data transform/approximation techniques include the discrete Fourier transform (DFT) [17], the discrete wavelet transform (DWT) [18,19], and piece-wise aggregate approximation (PAA) [20,21]. The DFT or DWT are used to transform data to their frequency domains, allowing feature extraction [22], and PAA is used as a data approximation approach. In data intensive fields, data transformations/approximations are commonly used as dimensionality reduction approaches, for obtaining fast approximate solutions for a given problem. Due to the ordered nature of genetic data, many of these transformation approaches can be applied to sequences of nucleotides [23] or amino acids [24]. An example of a successful implementation of a Fourier transform in computational biology is the alignment algorithm MAFFT [25] where the physiochemical properties of amino acids are used to represent sequences for fast matching of homologous sequence regions for alignment. Since most transformation approaches are suitable only for numerical sequences, the strings of letters representing genetic sequences must be mapped into numerical space using a numerical sequence representation method [26].

In addition to the DFT, the DWT and PAA, suitable methods for measuring the pairwise similarity of sequential data or transformations include the Lp-norms [27], dynamic time warping (DTW) [28], longest common subsequence (LCS) [29] and alignment approaches such as the Needleman-Wunsch and Smith-Waterman algorithms. Euclidean distance is arguably the most

widely used Lp-norm method for sequential data comparison but can only be used on sequences with same lengths. Furthermore, Lp-norm methods do not accommodate shifts in the x-axis (time or position) and are thus limited in their ability to identify similar features within offset data. Elastic similarity/dissimilarity methods such as LCS, unbounded DTW and various alignment algorithms permit comparison of data with different dimensions and tolerate shifts in the x-axis. These properties of elastic similarity methods can be very useful in the analysis of speech signals, for example, but can be computationally expensive [30]. Several approaches have been proposed to permit fast search with DTW, including the introduction of global constraints (wrapping path) or the use of lower bounding techniques such as LB_keogh [28].

While pairwise comparison methods may be used for clustering, classification and similarity searches, they are very time consuming for large datasets (O(n2) time complexity). Indexing structures such as the R*-tree, KD-tree, VP-tree and MVP-tree have significantly lower time complexity (O(n log(n))) for similarity search [31] and are more appropriate for efficient analysis of large datasets. The R*-tree [32,33] and KD-tree [34] indexing structures are very accurate for low dimensional datasets. However, their performance deteriorates significantly in high dimensional space [31], known as the 'curse of dimensionality' [35,36]. Metric trees such as the VP-tree [37] and MVP-tree [38] are less prone to this limitation. Metric space indexing structures make use of geometric properties for partitioning data and work efficiently on both low and high dimensional data [39]. The curse of dimensionality can be further mitigated using data approximations such as the DFT, the DWT, and the PAA to partition a dataset in an approximated space without loss of generality [21].

Here we investigate the performance of three established dimensionality reduction techniques, on three common analysis tasks involving viral short read sequence data: classification, reference based mapping/alignment, and *de novo* assembly. We benchmarked the accuracy of our proposed methodology against existing tools, and demonstrated the applicability of time series and signal processing data mining techniques for the analysis of viral NGS data.

2. Materials and Methods

2.1. Symbolic to numeric sequence representations

Various numeric sequence representation methods can be used for symbolising a nucleotide sequence to a numerical space (see Table 1). Depending on the chosen numerical representation, each nucleotide is associated with a specific numerical value or vector. The specific values are assigned to the position of each nucleotide indicating the presence of a nucleotide at each sequence position (Equation 1). R_i is the indicator for a specific nucleotide in the i^{th} position of the sequence S with a length of n nucleotides. Values $v_1 \ldots v_5$ correspond to the numerical value or numerical vector associated with each nucleotide.

$$R_i = \left\{ \begin{array}{l} v_1 \; if \; S_i = A \\ v_2 \; if \; S_i = T \\ v_3 \; if \; S_i = C \\ v_4 \; if \; S_i = G \\ v_5 \; otherwise \end{array} \right. , \forall i \in S_n \quad (1)$$

Methods such as the electron-ion interaction pseudo potentials (EIIP) [40] and the atomic representation approach [41] aim to mimic the biochemical properties of nucleic acids, but introduce some mathematical bias that does not exist in reality [26]. Other methods, like the Voss indictor [42] and the Tetrahedron approach, do not introduce internucleotide mathematical bias, meaning the pairwise distances between each non-identical transformed nucleotide is the same (for example, the distance between A and T is equal to the distances between A and C as well as A and G).

Furthermore, the cumulative sum of a numerical representation R can be used to indicate the trajectory of a sequence in nucleotide space. Table 1 indicates the associate values used for different representation methods [26].

2.2. Sequence Transformation

Effective methods for transforming/approximating sequential data should: *i)* accurately transform/approximate data without loss of useful information, *ii)* have low computational overheads, *iii)* facilitate rapid comparison of data, and *iv)* provide lower bounding—where the distance between data representations is always smaller than or equal to that of the original data—guaranteeing against false negative results [43]. We employ the DFT and the DWT transformation methods and PAA approximation method as they satisfy the above requirements, are widely used for analysing discrete signals [44], and can be used to transform/approximate nucleotide sequence numerical representations to different levels of resolution, permitting reduced dimensionality sequence analysis.

Figure 1A illustrates an example of the DFT and DWT transformations and PAA approximation of a short nucleotide sequence. DFT and the fast Fourier transform (FFT) transform data from their original domain to a frequency domain. In principle, the DFT decomposes a numerically represented nucleotide sequence with n positions (dimensions) into a series of n frequency components ordered by their frequency. A subset of the resulting Fourier frequencies are used to approximate the original sequence in a lower dimensional space [17], and the tradeoff between analytical speed and accuracy can be varied according to the number of frequencies considered [45]

DWT transform data from their original domain to their time-frequency, accommodating for changes in signal frequency over time [18,46,47]. DWT is a set of averaging and differencing functions that may be used recursively to represent sequential data at different resolutions and each resolution can be used as an approximation of the original data. Figure 1B depicts an example of the DWT transformations of a short nucleotide sequence.

In PAA a numerical sequence is divided into n equally sized windows, the mean values of which together form a compressed sequence representation [20,21]. The selection of n determines the resolution of the compressed or approximate representation. While PAA is faster and easier to implement than the DFT and the DWT, unlike these two methods it is irreversible, meaning that the original sequence cannot be recovered from the approximation. Figure 1C depicts an example of the PAA transformations of a short nucleotide sequence.

2.3. Similarity search approaches for sequential data

Here we adopt the Euclidian distance and *VP*-indexing tree to partition our data based on their approximate space distances and perform a fast *k*-nearest neighbor (*k*-NN) similarity search for aligning the reads to the reference genome.

In a VP-tree indexing structure, data partitioning is implemented in metric space. A data point which is used as a vantage point is selected (either randomly or by applying some heuristic to find and use the furthest point in the dataset [37]), and the rest of the data points are partitioned into two nodes based on their distance to that point. Data found to be closer to the vantage point than a given threshold (the median distance between all the data points and the vantage point) are assigned to the same node, and the rest of the data points to a different node. This function is repeated recursively in order to complete the partitioning process. The resulting indexing structure can then be used for fast identification of a *k*-nearest neighbour (*k*-NN) search. A *k*-NN-search returns the data points that are closest to a query *q*. Initially, the distance between the query q and the vantage point in the top node is calculated. If the distance between q and the vantage point satisfies a set of given conditions (the distance is smaller or larger than a given threshold – this threshold being the median distance between the vantage point and other data points within the node), a decision is made to visit either one or both of the nodes. This process is repeated until the entire tree has been traversed. The *k* data points—in this case reads— found closest to our query are the *k*-nearest neighbours to the query *q*.

2.4. Proposed short reads processing methodology

Our methodology for taxonomic classification, reference based mapping and *de novo* assembly of short reads, used time series and digital signal processing data transformation techniques. Figure 2 illustrates the fundamental concept of our approach. The short reads and reference genomes are mapped to a numerical space using an appropriate method from table 1. Subsequently, lower dimensional approximations are generated for all data using the appropriate data transformation method, such as DFT, DWT, and PAA. A VP-tree is constructed to allow fast data comparison. Depending on the application, the VP-tree is constructed either by using *k*-mer transformations obtained from the reference genomes or by using the short reads' transformations. Consequently, the best matches for our short reads' transformations are identified using a *k*-NN search approach on the VP-tree. As a final step, the results obtained from the *k*-NN search are re-evaluated in the original space to remove potential false positive results.

2.5. Data

The implementations of our proposed methodologies were assessed with both simulated and real virus datasets. The simulated datasets were generated using CuReSim [48] and WGSIM (https://github.com/lh3/wgsim). Simulated data included information such as the reference genome used, the alignment position, and alignment direction for each read, enabling rigorous evaluation of the proposed techniques. We used two simulators to generate different datasets to test our approach under every eventuality. CuReSim can generate long Ion Torrent reads, allowing the user to control the type of variation (insertion, deletion and substitution) to simulate. The user can also define the extent of variation for each individual type of variation. WGSIM can simulate diploid genomes with uniform insertion, deletion and substitution sequencing variation.

CuReSim was used to generate 16 HIV-HXB2 simulated datasets with different levels and types of variation. WGSIM was used to generate 4 mixed virus datasets with different levels of variation. Each simulation contained 200,000 reads generated using 5 Norovirus genomes, 5 Ebola virus genomes, and 5 Respiratory syncytial virus (RSV) genomes, with various types and extents of simulated variation. Table 2 contains detailed information about the simulated datasets.

Furthermore, 15 publicly available real virus datasets were used for the evaluation of our methodology. The real datasets comprise 5 Norovirus, 5 Ebola virus, and 5 human respiratory syncytial virus (RSV) short read datasets. Norovirus NGS datasets (ERR225628, ERR225629, ERR225631, ERR225632, ERR225633) were generated from diarrhoeal patients in Vietnam [49]. Group A rotavirus datasets were obtained from human and pig samples from Vietnam [50]. Human coronavirus NL63 datasets were obtained from Kenya [51]. The Ebola virus datasets (SRR3107337, SRR3107338, SRR3107340, SRR3107342, SRR3107343) were retrieved from the bioproject PRJNA309162, generated during the outbreaks in West Africa in 2013-2016 [52]. The human respiratory syncytial virus (RSV) datasets, ERR303259, ERR303260, ERR303261, ERR303262, ERR303263 [53], were generated from humans in Kenya. Further information for the real data can be found in Table 3.

The HIV-HXB2 genome (K03455) was used as a reference index to align and or run the taxonomic classification analysis for the HIV-HXB2 simulated dataset. The Norovirus genome KM198509, the Ebola virus genome KM034562, and the RSV genome KP317934, were used as a reference index to align and or run the taxonomic classification analysis for the mixed virus datasets. The Norovirus genome KM198509 was used to run the taxonomic classification analysis on the real Norovirus datasets, the Ebola virus genome KM034562 was used to run the taxonomic classification analysis on the real Ebola datasets, and the RSV genome KP317934, was used to perform the taxonomic classification analysis on the real RSV datasets. Further information for the reference genomes can be found in Table 4.

2.6. Classification and alignment evaluation

The accuracy of a classification and an alignment tool can be quantified in terms of F-measure [48], a balanced measure of precision and recall. With precision = true positive/true positive + false positive and recall = true positive/true positive + false negative, then F-measure = 2 x (precision x recall)/ (precision+recall) [48]. In the case of simulated data, information concerning the position of the read on the reference and alignment direction can be used to establish the correctness of an alignment, and thereby provide a more informative F-measure score. Unclassified reads are considered a false negative result. Any hit on the correct section of the genome and in the correct direction report is considered a true positive result. However, if the alignment position and direction information are not available, the F-measure can be calculated from the number of hits reported for a read, or the absence of a hit. Again, unclassified reads are considered false negative results, and classified reads are considered true positive results. In the case of mixed genome data, the F-measure score can be calculated by taking in consideration the number of hits that were reported for a read, as well as if a read was assigned to a reference genome from the same family. If a read is assigned to a genome from a different virus family, it is considered a false positive result, and unclassified reads are considered a false negative result.

3. Results

3.1. Classification by numbers (CBN)

For the taxonomic classification analysis, a classification tool was implemented in C++ (https://github.com/Avramis/ClassificationByNumber). The implementation was developed to evaluate our methodology but is not optimised for speed. Users may specify parameters such as the representation method, transformation method, search stringency and the *k*-mer length. A vantage-point tree (VP-Tree) indexing structure, containing information from a set of given genomic references, is used to classify reads. The initial step to build the VP-Tree is to extract all of the unique *k*-mers, of a user-specified size *k*, from a set of given reference genomes. Each unique *k*-mer is represented in numerical sequence, and then transformed to a lower dimensional space. The transformed data are then used to generate the VP-Tree indexing structure. Subsequently, each short read from a query set is converted into numerical space and transformed to a lower dimensional space and evaluated against the VP-tree. The approximate solution is then evaluated using the original data to identify false positive results. The CBN algorithm generates two output files. The first output is a text file providing detailed information on all of the classification hits generated for each read. The information generated for each classification includes the reference name, the direction in which the query read was aligned to the reference, the start and end position of the query on the refence, the alignment score, the CIGAR string, and the actual alignment of the query read on the refence genome. The second output file provides a brief review of the alignment. In every line, it contains the short read's name, the number of classifications generated for that particular read, the highest classification score obtained, the name of the reference which provided the highest classification score, the alignment direction and starting position on the reference.

The CBN tool was evaluated against NCBI-BLAST 2.8.1 BLASTn [54] and Kaiju 1.6.3 [55] classifier tools. BLASTn performs the analysis in nucleotide space, whereas Kaiju translates nucleotide sequences into every possible protein frame and performs the analysis in protein space. Figures 3, 4, and 5 illustrate the results of the classification evaluation process. Figure 3 shows the results obtained from the classification process on the HIV-HXB2 data. Figure 4 illustrates the results of the mixed virus datasets, and Figure 5 illustrates the results obtained from the real data.

In the taxonomic classification of HIV-HXB2 simulated data, where the short reads were classified against the genome that was used to generate them, Kaiju reported the highest accuracy scores. CBN outperformed BLASTn in most cases and only following behind in on the high variation rates datasets. For the mixed viruses simulated datasets, where reads where classified against species strains related to the ones used to generate the reads, BLASTn manage to correctly classify data to the correct species, followed closely by CBN, and Kaiju falling last. In the evaluation of the tools on the real data, where reads where classified against genome strains related to their respecting

species, CBN generated more accurate results compared to the other tools, followed by Kaiju, and BLASTn in third.

3.2. Alignment by numbers (ALBN)

To test the applicability of sequential data transformations and feature selection for read alignment, we implemented a prototype *k*-NN read aligner (Table 5) in C++ (available at https://github.com/Avramis/Alignment_by_numbers). As with the CBN classification analysis, the ALBN code was not optimised for speed, and users may specify parameters such as the representation method, transformation method, search stringency and the *k*-mer length used for seeding alignments. The algorithm's output was used to construct gapped alignments in the widely used Sequence Alignment/Map (SAM) file format.

The ALBN tool was evaluated against a set of well established, widely used, stated of the art tools such as, bowtie2 (version 2.3.1) [56], BWA-MEM (version0.7.16) [7], Graphmap (version 0.5.2) [57], and Segmehl (version 0.3.4) [58]. Each aligner's accuracy was quantified in terms of F-measure [48]. CuReSim provides information such as the simulated read's origin on the reference genome and its alignment direction, enabling evaluation of each aligner's output an calculation of alignment accuracy in terms of F-measure. For mixed virus datasets, tool performance was evaluated in terms of ability to match and align reads to the appropriate virus reference genome. For the real data, F-measures were calculated according to the number of reads aligned to the given genome or otherwise.

Figures 6, 7 and 8 illustrate the F-measures obtained by evaluating alignments from each aligner. Figure 6 illustrates alignment performance for each of the 16 datasets simulated using the K03455 HIV-HXB2 reference genome. Figure 7 illustrates alignment performance for virus reads simulated with Norovirus genome KM198509.1, Ebola genome KM034562.1, and the RSV genome KP317934.1. Figure 8-i to 8-iii illustrates alignment performance (F-measure) for alignments of real Norovirus, Ebola virus and RSV sequences against the same corresponding reference genomes used in the simulations.

ALBN provided highly accurate results in all data cases. Regarding the HIV-HXB2 data, where the short reads were aligned to genome that was used to generate them, ALBN provided the most accurate results in all 16 cases, followed by Bowtie2 in terms of accuracy. Also, in the case of the mixed virus datasets, where reads were aligned against genome strains related to those used to generate the dataset, ALBN provided the most accurate results, with Graphmap and BWA-MEM third and fourth respectively. ALBN also generated the most accurate alignment results using real data, where reads where aligned against a species-specific reference genome.

3.3. De novo assembly by numbers

Lastly, to test the applicability of this approach to the *de novo* assembly of short reads, we implemented a naïve algorithm for all-against-all *k*-mer comparison using data transformations/approximation. Figure 9 illustrates the main concept of our *de novo* assembly approach. For the ASBN tool, reads are represented as numerical sequences using an appropriate numerical representation method (Table 1). Here we use the tetrahedron numerical representation approach. Every *k*-mer of each numerically represented read is identified and transformed to lower dimensional space using the chosen transformation method. All *k*-mers' transformations are used to build a VP-tree, to allow for fast data comparison. Afterwards, all *k*-mers are compared to the rest of the data using the VP-tree index. Information from the data comparison is used to construct a weighted graph similar to that shown in Figure 9A. The shortest path on the weighted graph is identified with a breadth-first search (BFS) (Figure 9B). Reads overlaps are used to generate an OLC alignment of short reads (Figure 9C).

The ASBN assembler was compared with Megahit (version 1.1.3) [59], and SPAdes (version 3.13.0) [60] *de novo* assemblers on the HIV-HXB2 and mixed virus simulated datasets accordingly. The derived contigs from each assembler were evaluated against the reference genomes used to generate the data simulations with BLASTn [54]. Contig positions across the genome were collected

and plotted in a histogram. Secondly, a measure of assembly progress was plotted on an X-Y matrix with X being the total coverage of the genomes generated and Y being the total number of gaps in the coverage. A perfect assembly would have X = full genome length and Y = 0, implicating, that the contig matches the genome in terms of size and nucleotide sequences, without any mismatches or gaps. For the HIV-HXB2 datasets, the contigs were evaluated against the K03455 genome, and the contigs obtained from the mixed virus datasets were evaluated against the 15 different genomes, KM198529, KM198528, KM198511, KM198500, KM198486, KU296608, KU296553, KU296549, KU296528, KU296416, KP317952, KP317946, KP317934, KP317923, and KP317922.

Figure 10 illustrates the assembly results of SPAdes, Megahit and all three variants of ASBN on the 16 simulated HIV-HXB2 datasets, and Figure 11 illustrates the assembly results on the mixed virus simulated databases. Although ASBN processes data and assembles short reads in a lower dimensional space, it nevertheless generated contigs that collectively cover the expected genome length and provided comparable results to both state of the art *de novo* assemblers in this experiment(Figure 10, Figure 11). In all cases, ASBN generated contigs spanning the whole genomes of their respective viral species.

4. Discussion

Although well-established data compression methods for reversible compression of one-dimensional and multivariate signals, images, text and binary exist [61-63]; there are very few examples of their application to biological sequence data. We developed algorithms incorporating signal compression methods for three common biological sequence analysis problems: classification, alignment and *de novo* assembly of NGS short read virus data. Our results show that our approach permits accurate classification and reference alignment in spite of high rates of sequence variation or the use of a divergent reference genome. Data approximation/summarisation techniques such as the DFT, the DWT and the PAA can be used to extract major features of sequence data and to suppress noise or low-level variation. This allows data comparison in terms of major characteristics, thus enabling the identification of similarities among data that might otherwise be concealed by minor variation or noise.

Collectively these results demonstrate that complete nucleotide-level sequence resolution is not a prerequisite of accurate sequence analysis, and that analytical performance can be preserved or even enhanced through appropriate dimensionality reduction (compression) of sequences. While our implementations use *k*-mers, the nature of the transformation/compression methods used showed that optimal *k*-mer selection is far less important than with conventional exact *k*-mer matching methods. The inherent error tolerance of the approach also permits use of larger k values than are normally used with conventional sequence comparison algorithms, reducing the computational burden of pairwise comparison, and thus, in *de novo* assembly specifically, the complexity of building and searching an assembly graph.

Efficient mining of terabase-scale biological sequence datasets requires looking beyond substring-indexing algorithms towards more versatile methods of compression for both data storage and analysis. The use of probabilistic data structures can considerably reduce the computer memory required for in-memory sequence lookups at the expense of a few false positives, and Bloom filters and related data structures have seen broad application in *k*-mer centric tasks such as error correction [64], *in silico* read normalisation [65] and *de novo* assembly [66,67]. However, while these hash-based approaches perform well on datasets with high sequence redundancy, for large datasets with many distinct *k*-mers, large amounts of memory are still necessary [65]. Lower bounding transformations and approximation methods (such as the DFT, the DWT and PAA) can exhibit the same attractive one-sided error offered by these probabilistic data structures, but instead of hash tables use concrete and thus reusable sequence representations.

Furthermore, transformations allow compression of standalone sequence composition, enabling flexible reduction of sequence resolution according to analytical requirements, so that redundant sequence precision need not hinder analysis. While the problem of read alignment to a known reference sequence is largely considered solved, assembly of large genomes remains a

formidable problem in computing. Moreover, consideration of the metagenomic composition of mixed biological samples, as demonstrated, further extends the scope and scale of the assembly problem beyond what is tractable using conventional sequence comparison approaches. By implementing a reference-based aligner and *de novo* assembler, we have demonstrated that using compressed numerical representations representa a tractable and versatile approach for reconstructing genomes and metagenomes sequenced with short reads.

In conclusion, short nucleotide sequences may be effectively represented as numerical series, enabling the application of existing analytical methods from a variety of mathematical and engineering fields for the purposes of sequence alignment and assembly. By applying established signal decomposition methods, compressed representations of nucleotide sequences can be created, permitting reductions in the spatiotemporal complexity of their analysis, without necessarily compromising analytical accuracy.

Authors' Contribution

AT designed and wrote the methods and software, and performed the data analysis with help from BC, MP, SK, and MC. BC and MC, generated the simulated data and BC, MP and MC help on the data evaluation. AT and DLR conceived the study. AT and BC wrote the manuscript with comments from MP, MC and DLR. All authors read and approved the final manuscript.

Funding

This work has been supported by the Wellcome Trust [097820/Z/11/B]; the BBSRC [BB/H012419/1 and BB/M001121/1 and BC by a BBSRC DTP studentship to DLR]; and the VIROGENESIS project which receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 634650.

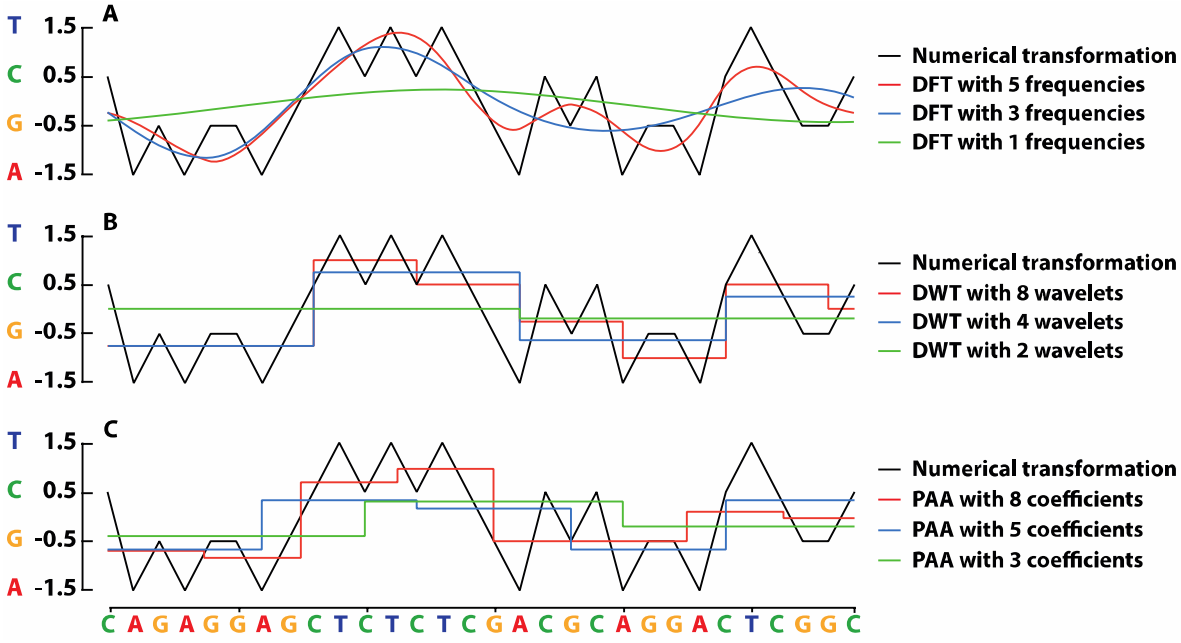

Figure 1. A numerically represented DNA sequence transformed at various levels of spatial resolution using the discrete Fourier transform (DFT) of the whole sequence (A), the Haar discrete wavelet transform (DWT) (B), and piecewise aggregate approximation (PAA) (C). A 30 nucleotide sequence (x-axis) is represented as a numerical sequence (black lines) using the real number representation method (y-axis where T=1.5, C=0.5, G=-0.5 and A=-1.5) for DFT approximations of the sequence with 5 (red), 3 (blue) and 1 (green) Fourier frequencies (A); DWT approximations of the

same sequence with 8 level wavelets (red), 4 level wavelets (blue), and 2 level wavelets (green) (B); and PAA approximations of the same sequence with 8 (red), 5 (blue) and 3 (green) coefficients (C).

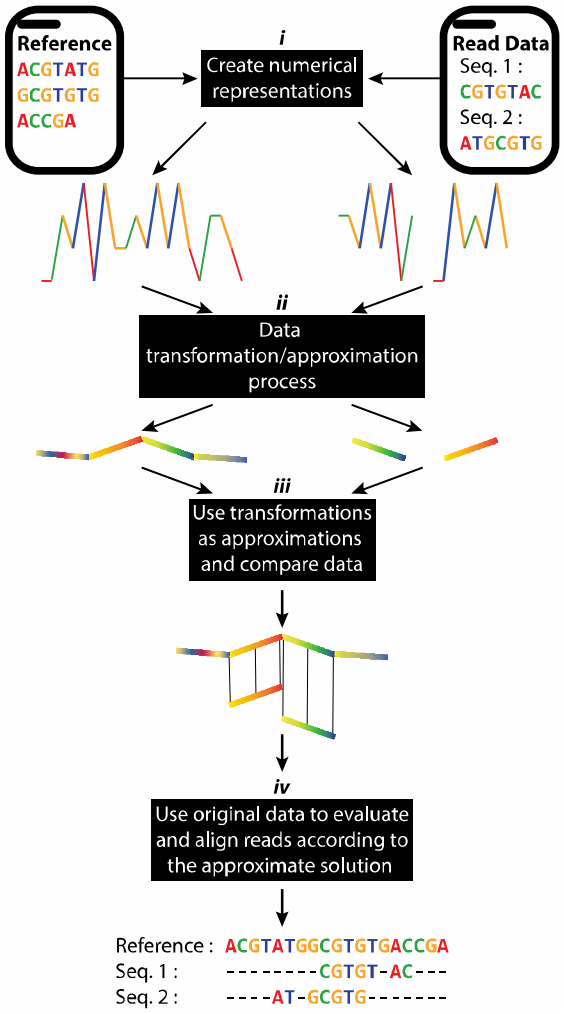

Figure 2. Overview of our proposed methodology using time series transformation/approximation methods: (i) Creation of numerical representations of input sequences. (ii) Application of an appropriate signal decomposition method to transform sequences into their feature space. (iii) Use of approximated transformations to perform rapid data analysis in lower dimensional space. (iv) Validation of inferences against original, full-resolution input sequences. In the case of reference-based alignment, and taxonomic classification, approximated read transformations were compared with a reference sequence. In our *de novo* implementation, pairwise comparisons were performed between all of the approximated read transformations.

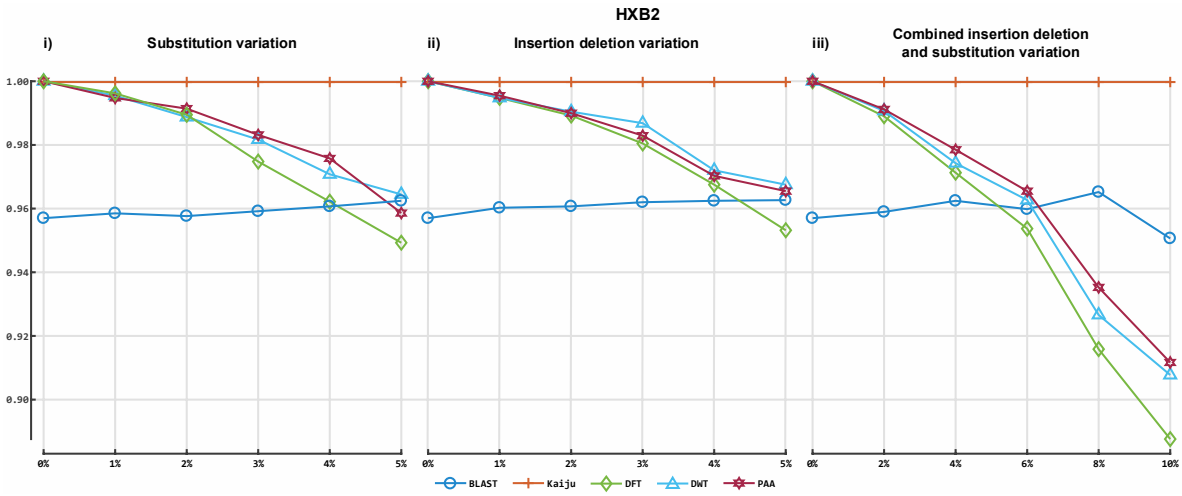

Figure 3. Accuracy of our prototype classification implementation and two established tools on HIV-HXB2 simulated datasets. All plots illustrate the F-measures obtained on the 16 different HIV datasets. The Y axis indicates the F-measure score, and the X axis depicts the reads data files. Plot 3-i depicts the F-measures obtained for each classifier on the simulations with 0% to 5% of substitution variation rate. Plot 6-ii illustrates the F-measures obtained for each classifier on the simulations with 0% to 5% uniform insertion/deletion variation and plot 3-iii illustrates the F-measures obtained for each tool on simulations of uniform 0% to 10% insertion/deletion and substitution variation.

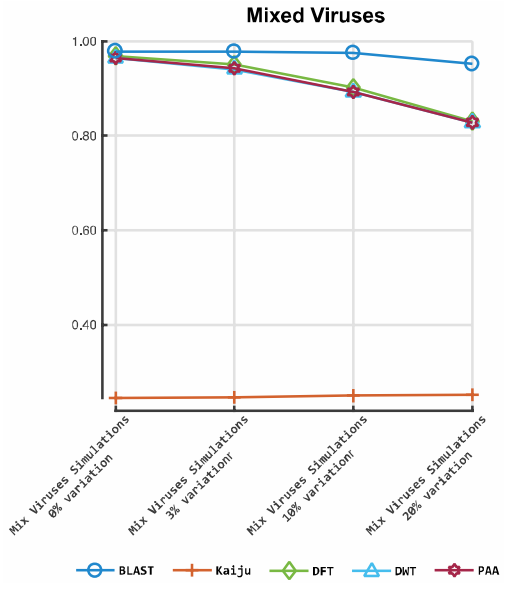

Figure 4. Accuracy of our prototype classification implementation and two established tools on mixed viruses simulated datasets. The Y axis indicates the F-measure score, and the X axis depicts the reads data files. The plot depicts the F-measures obtained for each classifier on the mixed virus simulations.

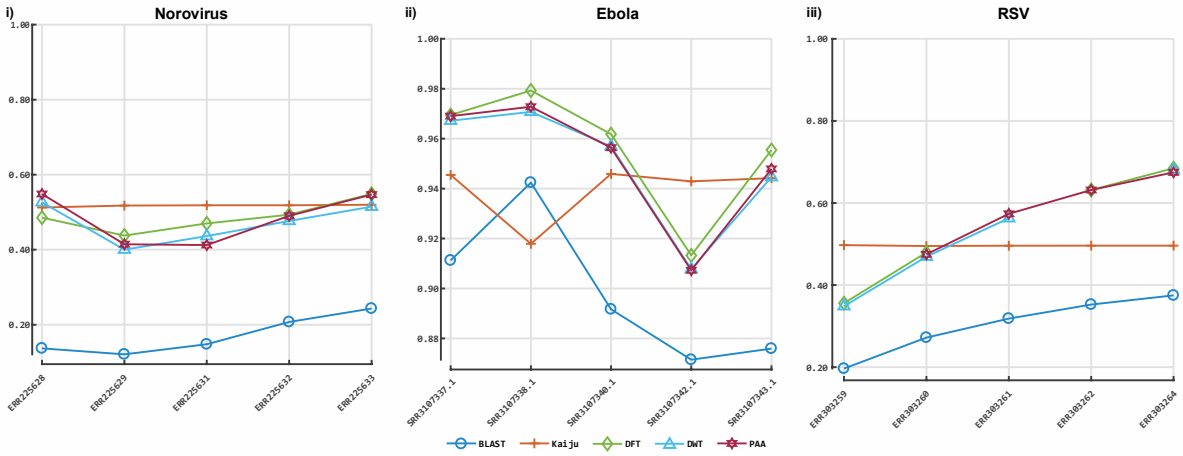

Figure 5. Accuracy of our prototype classification implementation and two established tools on real sequences. The Y axis indicates the F-measure score, and the X axis depicts the reads data files. The Y axis indicates the F-measure score, and the X axis depicts the reads data files. Plot 5-i depicts the F-measures obtained for each classifier on the Norovirus sequences data. Plot 5-ii illustrates the F-measures obtained for each classifier on the Ebola sequence data. Plot 5-iii illustrates the F-measures obtained for each tool on Respiratory syncytial virus (RSV) sequence data.

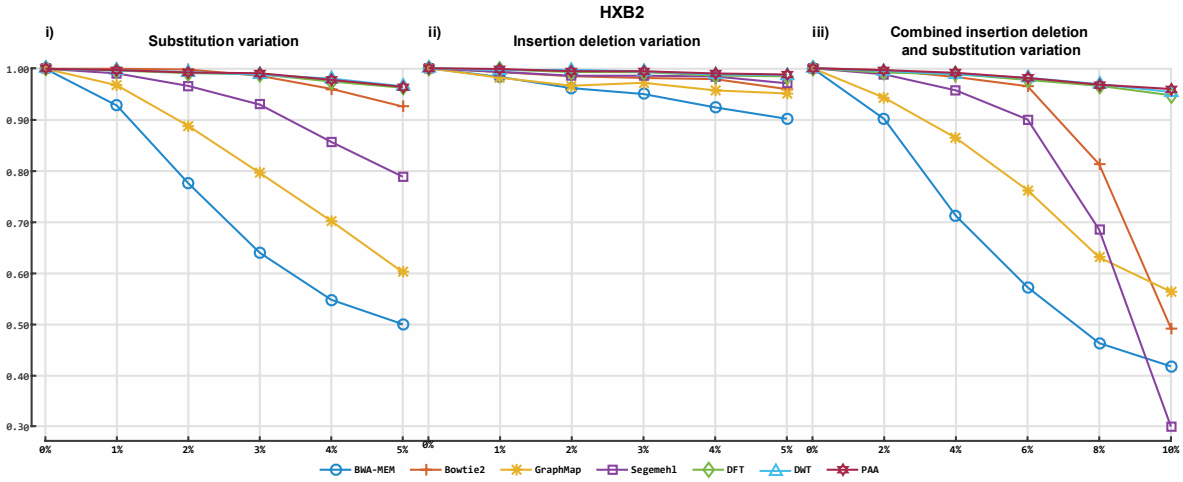

Figure 6. Accuracy of our prototype reference alignment implementation and four established tools on HIV-HXB2 simulated datasets. Figure 6 illustrates the F-measures obtained on the 16 different HIV datasets. Plot 6-i depicts the F-measures obtained for each aligner on the simulations with 0% to 5% of substitution variation rate. Plot 6-ii illustrates the F-measures obtained for each aligner on the simulations with 0% to 5% uniform insertion/deletion variation and plot 6-iii illustrates the F-measures obtained for each tool on simulations of uniform 0% to 10% insertion/deletion and substitution variation.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 April 2019 doi:10.20944/preprints201904.0014.v1

Peer-reviewed version available at Viruses 2019, 11, 394; doi:10.3390/v11050394

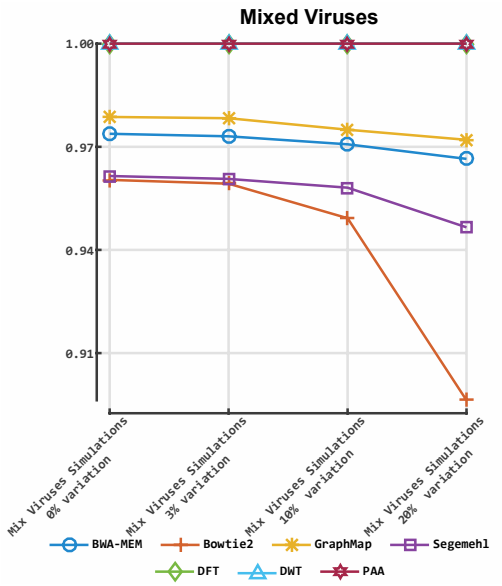

Figure 7. Accuracy of our prototype aligner implementation and four established tools on mixed viruses simulated datasets. The Y axis indicates the F-measure score, and the X axis depicts the reads data files. The plot depicts the F-measures obtained for each aligner on the mixed virus simulations

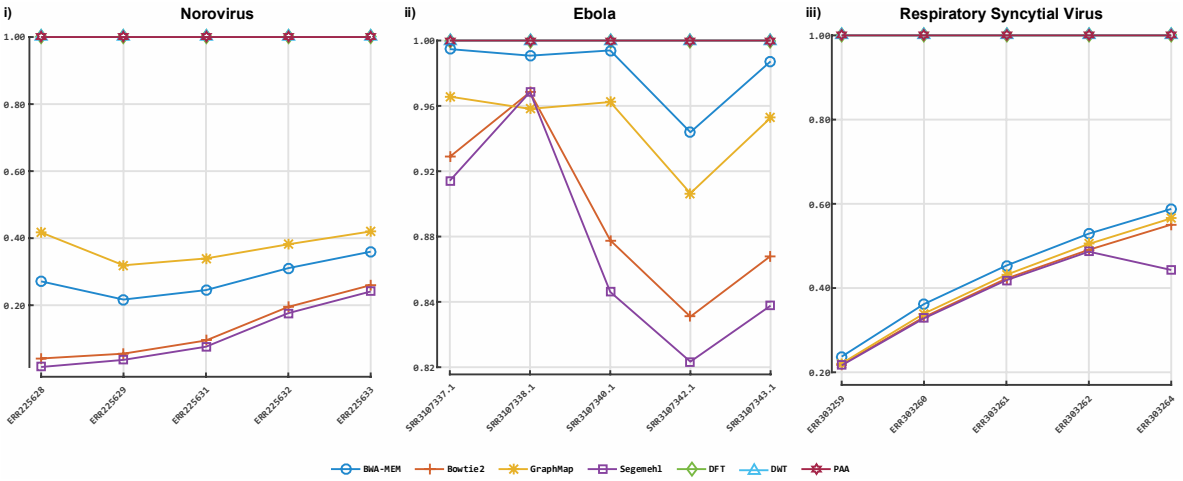

Figure 8. Accuracy of our prototype aligner implementation and four established tools on real sequences datasets. The Y axis indicates the F-measure score, and the X axis depicts the reads data files. The Y axis indicates the F-measure score, and the X axis depicts the reads data files. Plot 8-i depicts the F-measures obtained for each aligner on the Norovirus sequences data. Plot 8-ii illustrates the F-measures obtained for each aligner on the Ebola sequences data. Plot 8-iii illustrates the F-measures obtained for each tool on the Respiratory syncytial virus (RSV) sequences data.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 April 2019 doi:10.20944/preprints201904.0014.v1

Peer-reviewed version available at Viruses 2019, 11, 394; doi:10.3390/v11050394

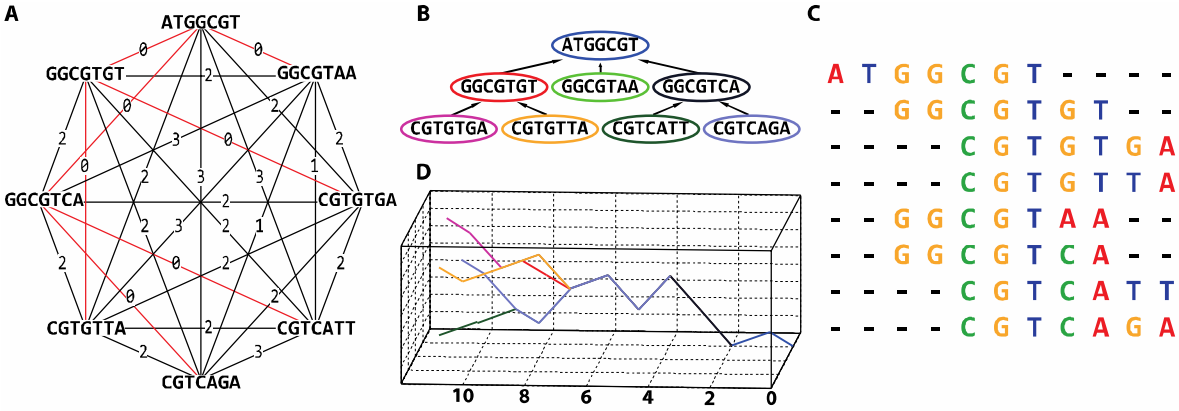

Figure 9. A *de novo* assembly methodology for numerically represented nucleotide reads. All-against-all sequence comparison (A) enables construction of a read graph with weighted edges. The weight assigned to each edge is the smallest pairwise distance between every possible *k*-mer representation of the two reads. (B) The shortest path in the graph is identified with a breadth-first search algorithm (red coloured edges) thereby (C) enabling read alignment. A DNA walk representation of the overlapped reads (D) may subsequently be used as a three-dimensional graphical portrayal of the reads, illustrating alignment characteristics.

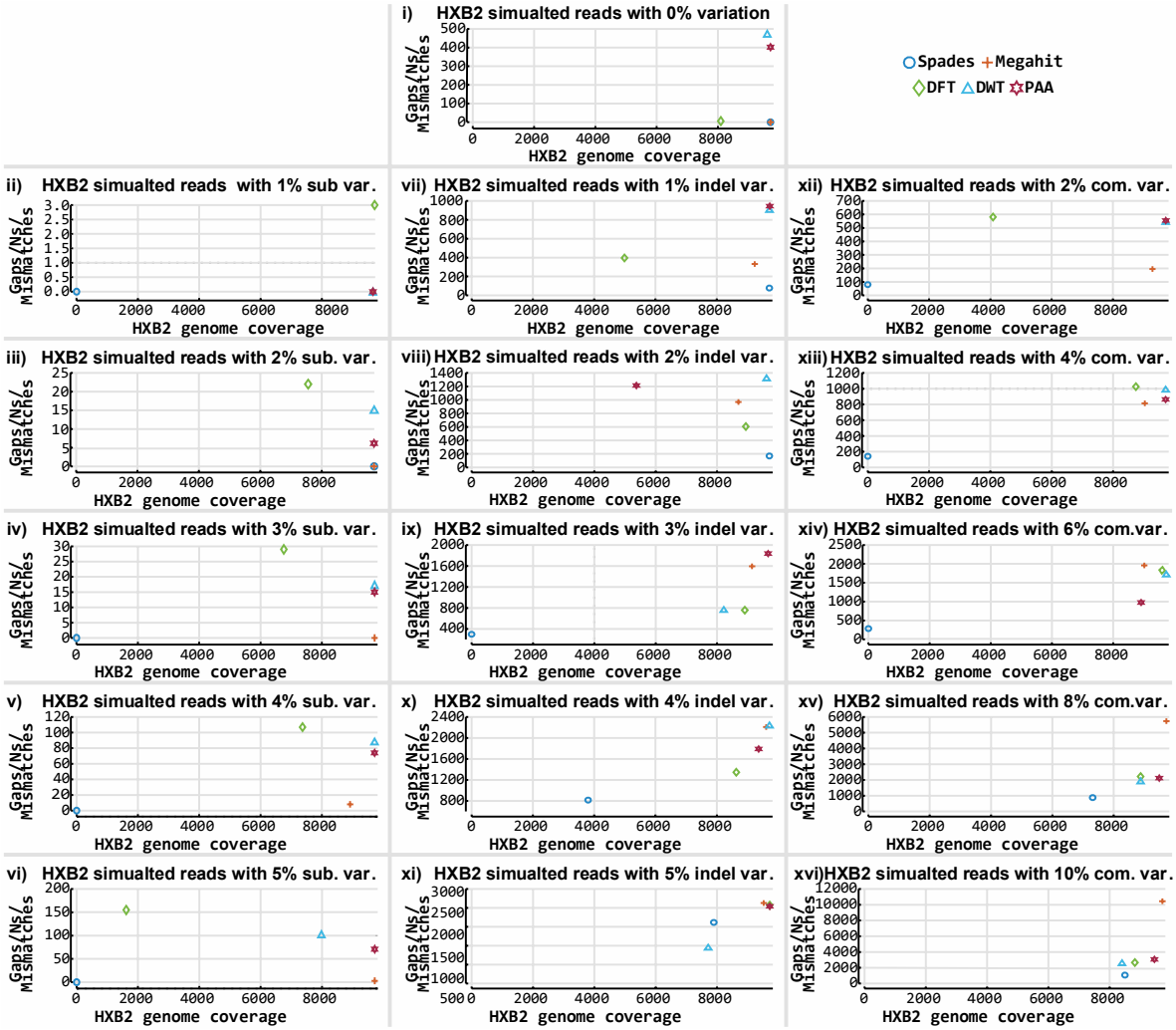

Figure 10. Accuracy of our prototype *de novo* assembly implementation and two established tools on HIV-HXB2 simulated datasets. The contigs obtained for each assembler were evaluated against the reference genome used to generate the simulated data. BLASTn was used to align all contigs to a

HIV-HXB2 refence genome and determine genome coverage. The Y axis indicates the number of gaps and mismatches that exist in the contigs obtained for each tool, and the X axis depicts the length of the genome the reported contigs cover. The contigs obtained from the assembly of the HIV-HXB2 simulated short read data were evaluated against the K03455 reference genome. Plot 10-i illustrates results obtained from all assemblers on variation free data. Plots 10-ii to 10-vi illustrate results obtained from all assemblers on data with different levels of substitution variation. Plots 10-vii to 10-xi illustrate results obtained from all assemblers on data with different levels of insertion/deletion variation. Plots 10-xiii to 10-xvi illustrate results obtained from all assemblers on data with different levels of combined insertion/deletion and substitution variation.

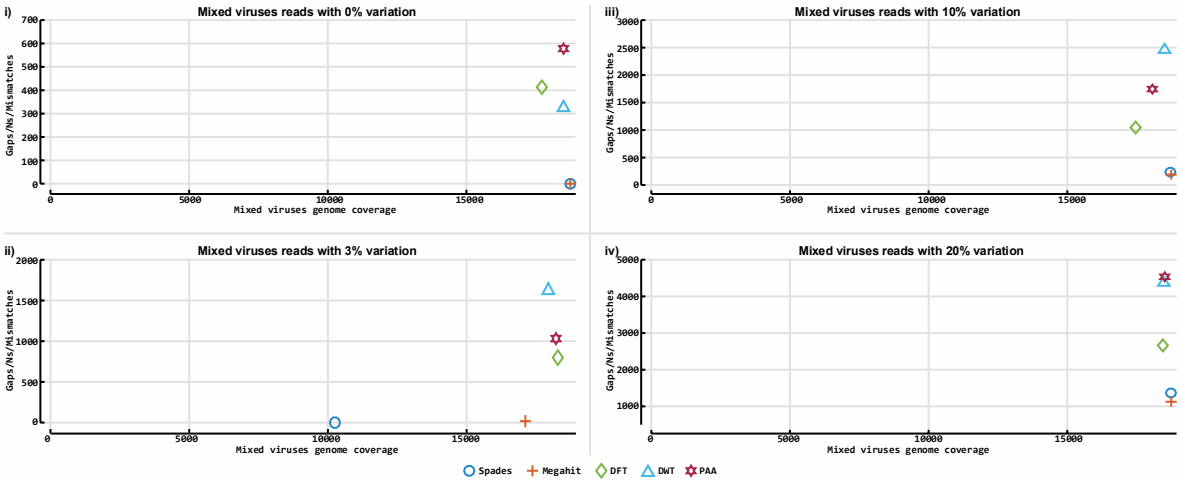

Figure 11. Accuracy of our prototype *de novo* assembly implementation and two established tools on mixed viruses simulated datasets**.** The contigs obtained for each assembler were evaluated against the reference genome that was used to generate the simulated data. BLASTn was used to align all contigs to a HIV-HXB2 refence genome, and determine how much of the particular genome they do cover. The Y axis indicates the number of gaps and mismatches that exist in the contigs obtained for each tool, and the X axis depicts the length of the genome the reported contigs cover. The contigs obtained from the mixed virus simulated dataset were evaluated against the, KM198529, KM198528, KM198511, KM198500, KM198486, KU296608, KU296553, KU296549, KU296528, KU296416, KP317952, KP317946, KP317934, KP317923, and KP317922 references genomes. Plots 11-i to 11-iv illustrate results obtained from all assemblers on data with 0%, 3%, 10% and 20% variation levels accordingly.

503 **Table 1** Numerical nucleotide sequences representation methods.

Integer number	$A=1,\ C=-1,\ G=2,\ T=-2,\ N=0$
Real number	$A=-1.5,\ C=0.5,\ G=-0.5,\ T=1.5,\ N=0.0$
EIIP	$A=0.1260,\ C=0.1340,\ G=0.0806,\ T=0.1335,$ N=∅
Atomic	$A=70,\ C=58,\ G=78,\ T=66,$ N=∅
Pair	A *or* $T=1,\ C$ or $G=-1,$ N=0
Complex number	$A=1+1i,\ C=-1+1i,\ G=-i-1i,\ T=1-1i,$ N=0+0i
DNA Walk	$A=[1,0],\ C=[0,1],$ G$=[0,-1],\ T=[-1,0],\ N=[0,0]$
Tetrahedron	$A=[0,0,1],$ C$=\left[-\frac{\sqrt{2}}{3},\ -\frac{\sqrt{6}}{3},\ \frac{1}{3}\right],$ G$=\left[-\frac{\sqrt{2}}{3},\ -\frac{\sqrt{6}}{3},\ -\frac{1}{3}\right],$ T$=\left[2\times\frac{\sqrt{2}}{3},\ 0,\ -\frac{1}{3}\right],$ N$=[0,\ 0,\ 0]$
Voss indicator	$A=[0,0,1,0],\ C=[1,0,0,0],\ G=[0,1,0,0],$ $T=[0,0,0,1],\ N=[0,0,0,0]$

504

505

506

507 **Table 2.** Simulated read data. Each row contains details for each simulated dataset (i.e. virus family, virus,
508 GenBank id, variation type, variation level, number of reads, and simulator used to generate data).

Family	Virus	GenBank genome ID	Variation Type (%)			Reads	Simulator
			Ins	Del	Sub		
HIV	HXB2	K03455	0.0	0.0	0.0	2133	Curesim
HIV	HXB2	K03455	0.0	0.0	1.0	2133	Curesim
HIV	HXB2	K03455	0.0	0.0	2.0	2133	Curesim
HIV	HXB2	K03455	0.0	0.0	3.0	2133	Curesim
HIV	HXB2	K03455	0.0	0.0	4.0	2133	Curesim
HIV	HXB2	K03455	0.0	0.0	5.0	2133	Curesim
HIV	HXB2	K03455	0.5	0.5	0.0	2133	Curesim
HIV	HXB2	K03455	1.0	1.0	0.0	2133	Curesim
HIV	HXB2	K03455	1.5	1.5	0.0	2133	Curesim
HIV	HXB2	K03455	2.0	2.0	0.0	2133	Curesim
HIV	HXB2	K03455	2.5	2.5	0.0	2133	Curesim
HIV	HXB2	K03455	0.5	0.5	1.0	2133	Curesim
HIV	HXB2	K03455	1.0	1.0	2.0	2133	Curesim
HIV	HXB2	K03455	1.5	1.5	3.0	2133	Curesim
HIV	HXB2	K03455	2.0	2.0	4.0	2133	Curesim
HIV	HXB2	K03455	2.5	2.5	5.0	2133	Curesim
Mixed Viruses: Caliciviridae, Filoviridae, Pneumoviridae	Norovirus, Ebola virus, RSV	KM198529, KM198528, KM198511, KM198500, KM198486, KU296608, KU296553, KU296549, KU296528, KU296416, KP317952, KP317946, KP317934, KP317923, KP317922	0.0	0.0	0.0	200000	WGSIM
Mixed Viruses: Caliciviridae, Filoviridae, Pneumoviridae	Norovirus, Ebola virus, RSV	KM198529, KM198528, KM198511, KM198500, KM198486, KU296608, KU296553, KU296549, KU296528, KU296416, KP317952, KP317946, KP317934, KP317923, KP317922	1.0	1.0	1.0	200000	WGSIM
Mixed Viruses, Caliciviridae, Filoviridae, Pneumoviridae	Norovirus, Ebola virus, RSV	KM198529, KM198528, KM198511, KM198500, KM198486, KU296608, KU296553, KU296549, KU296528, KU296416, KP317952, KP317946, KP317934, KP317923, KP317922	3.33	3.33	3.33	100000	WGSIM
Mixed Viruses, Caliciviridae, Filoviridae, Pneumoviridae	Norovirus, Ebola virus, RSV	KM198529, KM198528, KM198511, KM198500, KM198486, KU296608, KU296553, KU296549, KU296528, KU296416, KP317952, KP317946, KP317934, KP317923, KP317922	6.66	6.66	6.66	200000	WGSIM

509

510 **Table 3.** Real short reads data. Rows contain information for each real reads' dataset (i.e. virus
511 family, virus, genome strain GenBank id, SRA project ID, number of reads, and technology used to
512 sequence data).

Family	Virus	Amplicon/random primer	GenBank genome ID	ENA/SRA_ID	Reads	Sequencing Technology
Caliciviridae	Norovirus	Amplicon	KM198486	ERR225628	2126502	Illumina MiSeq
Caliciviridae	Norovirus	Amplicon	KM198500	ERR225629	3037674	Illumina MiSeq
Caliciviridae	Norovirus	Amplicon	KM198511	ERR225631	3285078	Illumina MiSeq
Caliciviridae	Norovirus	Amplicon	KM198528	ERR225632	4361884	Illumina MiSeq
Caliciviridae	Norovirus	Amplicon	KM198529	ERR225633	5187234	Illumina MiSeq
Filoviridae	Ebola virus	Amplicon	KU296608	SRR3107337	522968	Ion Torrent PGM
Filoviridae	Ebola virus	Amplicon	KU296549	SRR3107338	771031	Ion Torrent PGM
Filoviridae	Ebola virus	Amplicon	KU296416	SRR3107340	186657	Ion Torrent PGM
Filoviridae	Ebola virus	Amplicon	KU296553	SRR3107342	478346	Ion Torrent PGM
Filoviridae	Ebola virus	Amplicon	KU296528	SRR3107343	42410	Ion Torrent PGM
Pneumoviridae	RSV	Amplicon	KP317934	ERR303259	7275032	Illumina MiSeq
Pneumoviridae	RSV	Amplicon	KP317922	ERR303260	9278070	Illumina MiSeq
Pneumoviridae	RSV	Amplicon	KP317946	ERR303261	11111114	Illumina MiSeq
Pneumoviridae	RSV	Amplicon	KP317923	ERR303262	13293226	Illumina MiSeq
Pneumoviridae	RSV	Amplicon	KP317952	ERR303263	15237848	Illumina MiSeq

513

514

Peer-reviewed version available at Viruses 2019, 11, 394; doi:10.3390/v11050394

Table 4. Reference genomes used during classification and reference based alignment

Family	Virus	GenBank id:	Length (nt)
Retroviridae	Human immunodeficiency virus 1 (HXB2)	K03455	9179
Caliciviridae	Norovirus	KM198509.1	7425
Filoviridae	Zaire ebolavirus	KM034562.1	18957
Pneumoviridae	Human orthopneumovirus (Respiratory Syncytial Virus)	KP317934.1	15233

Table 5. Pseudocode for the alignment procedure

1) Represent short reads and reference genome as numerical sequences.
2) Select *k*-mer length.
3) Create transformations of each reference sequence *k*-mer, build VP-tree, and create transformations of the initial *k*-mer of each short reads.
4) Identify candidate alignments using data transformations.

```
for each read i
    candidate_alignments[i] = VPtree.k-NNSearch(query i)
end
```

5) Align approximate results with original data using the Smith-Waterman (SW) algorithm:

```
for each read i
    best_score = null
   best_aln = []
    for each k neighbour in candidate_alignments[i]
        if SW_score(k neighbour, read i)
            best_score = SW_score(k neighbour, read i)
            best_aln = SW_aln(k neighbour, read i)
        end
    end
end
```

6) Output alignment in Sequence Alignment/Map (SAM) format.

References

1. Margulies, M.; Egholm, M.; Altman, W.E.; Attiya, S.; Bader, J.S.; Bemben, L.A.; Berka, J.; Braverman, M.S.; Chen, Y.-J.; Chen, Z. Genome sequencing in microfabricated high-density picolitre reactors. *Nature* **2005**, *437*, 376-380.
2. Bentley, D.R.; Balasubramanian, S.; Swerdlow, H.P.; Smith, G.P.; Milton, J.; Brown, C.G.; Hall, K.P.; Evers, D.J.; Barnes, C.L.; Bignell, H.R. Accurate whole human genome sequencing using reversible terminator chemistry. *Nature* **2008**, *456*, 53-59.

3. Rothberg, J.M.; Hinz, W.; Rearick, T.M.; Schultz, J.; Mileski, W.; Davey, M.; Leamon, J.H.; Johnson, K.; Milgrew, M.J.; Edwards, M. An integrated semiconductor device enabling non-optical genome sequencing. *Nature* **2011**, *475*, 348-352.
4. Eid, J.; Fehr, A.; Gray, J.; Luong, K.; Lyle, J.; Otto, G.; Peluso, P.; Rank, D.; Baybayan, P.; Bettman, B. Real-time DNA sequencing from single polymerase molecules. *Science* **2009**, *323*, 133-138.
5. Salipante, S.J.; Roach, D.J.; Kitzman, J.O.; Snyder, M.W.; Stackhouse, B.; Butler-Wu, S.M.; Lee, C.; Cookson, B.T.; Shendure, J. Large-scale genomic sequencing of extraintestinal pathogenic Escherichia coli strains. *Genome research* **2014**, gr. 180190.180114.
6. Rose, R.; Constantinides, B.; Tapinos, A.; Robertson, D.L.; Prosperi, M. Challenges in the analysis of viral metagenomes. *Virus Evolution* **2016**, *2*, vew022.
7. Li, H.; Durbin, R. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics* **2009**, *25*, 1754-1760.
8. Shrestha, A.M.S.; Frith, M.C.; Horton, P. A bioinformatician's guide to the forefront of suffix array construction algorithms. *Briefings in bioinformatics* **2014**, *15*, 138-154.
9. Myers, E.W. Toward simplifying and accurately formulating fragment assembly. *Journal of Computational Biology* **1995**, *2*, 275-290.
10. Kececioglu, J.D.; Myers, E.W. Combinatorial algorithms for DNA sequence assembly. *Algorithmica* **1995**, *13*, 7-51.
11. Earl, D.; Bradnam, K.; John, J.S.; Darling, A.; Lin, D.; Fass, J.; Yu, H.O.K.; Buffalo, V.; Zerbino, D.R.; Diekhans, M. Assemblathon 1: A competitive assessment of de novo short read assembly methods. *Genome research* **2011**, *21*, 2224-2241.
12. Iqbal, Z.; Caccamo, M.; Turner, I.; Flicek, P.; McVean, G. De novo assembly and genotyping of variants using colored de Bruijn graphs. *Nature genetics* **2012**, *44*, 226-232.
13. Pevzner, P.A.; Tang, H.; Waterman, M.S. An Eulerian path approach to DNA fragment assembly. *Proceedings of the National Academy of Sciences* **2001**, *98*, 9748-9753.
14. Bradnam, K.R.; Fass, J.N.; Alexandrov, A.; Baranay, P.; Bechner, M.; Birol, I.; Boisvert, S.; Chapman, J.A.; Chapuis, G.; Chikhi, R. Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. *GigaScience* **2013**, *2*, 1-31.
15. Archer, J.; Rambaut, A.; Taillon, B.E.; Harrigan, P.R.; Lewis, M.; Robertson, D.L. The evolutionary analysis of emerging low frequency HIV-1 CXCR4 using variants through time—an ultra-deep approach. *PLoS computational biology* **2010**, *6*, e1001022.
16. Clement, N.L.; Thompson, L.P.; Miranker, D.P. ADaM: augmenting existing approximate fast matching algorithms with efficient and exact range queries. *BMC bioinformatics* **2014**, *15*, S1.
17. Agrawal, R.; Faloutsos, C.; Swami, A. *Efficient similarity search in sequence databases*; Springer: 1993.
18. Chan, K.-P.; Fu, A.-C. Efficient time series matching by wavelets. In Proceedings of Data Engineering, 1999. Proceedings., 15th International Conference on; pp. 126-133.
19. Woodward, A.M.; Rowland, J.J.; Kell, D.B. Fast automatic registration of images using the phase of a complex wavelet transform: application to proteome gels. *Analyst* **2004**, *129*, 542-552.
20. Geurts, P. Pattern extraction for time series classification. In *Principles of Data Mining and Knowledge Discovery*, Springer: 2001; pp. 115-127.
21. Keogh, E.; Chakrabarti, K.; Pazzani, M.; Mehrotra, S. Locally adaptive dimensionality reduction for indexing large time series databases. *ACM SIGMOD Record* **2001**, *30*, 151-162.

22. Shumway, R.H.; Stoffer, D.S.; Stoffer, D.S. *Time series analysis and its applications*; Springer New York: 2000; Vol. 3.
23. Silverman, B.; Linsker, R. A measure of DNA periodicity. *Journal of theoretical biology* **1986**, *118*, 295-300.
24. Cheever, E.; Searls, D.; Karunaratne, W.; Overton, G. Using signal processing techniques for DNA sequence comparison. In Proceedings of Bioengineering Conference, 1989., Proceedings of the 1989 Fifteenth Annual Northeast; pp. 173-174.
25. Katoh, K.; Misawa, K.; Kuma, K.i.; Miyata, T. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic acids research* **2002**, *30*, 3059-3066.
26. Kwan, H.K.; Arniker, S.B. Numerical representation of DNA sequences. In Proceedings of Electro/Information Technology, 2009. eit'09. IEEE International Conference on; pp. 307-310.
27. Yi, B.-K.; Faloutsos, C. Fast time sequence indexing for arbitrary Lp norms.
28. Keogh, E.; Ratanamahatana, C.A. Exact indexing of dynamic time warping. *Knowledge and information systems* **2005**, *7*, 358-386.
29. Vlachos, M.; Kollios, G.; Gunopulos, D. Discovering similar multidimensional trajectories. In Proceedings of Data Engineering, 2002. Proceedings. 18th International Conference on; pp. 673-684.
30. Kotsakos, D.; Trajcevski, G.; Gunopulos, D.; Aggarwal, C.C. Time-Series Data Clustering. 2013.
31. Chávez, E.; Navarro, G.; Baeza-Yates, R.; Marroquín, J.L. Searching in metric spaces. *ACM computing surveys (CSUR)* **2001**, *33*, 273-321.
32. Beckmann, N.; Kriegel, H.-P.; Schneider, R.; Seeger, B. *The R*-tree: an efficient and robust access method for points and rectangles*; ACM: 1990; Vol. 19.
33. Lin, R.; King-lp, A.; Shim, H.S.S.K. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In Proceedings of Proceeding of the 21th International Conference on Very Large Data Bases; pp. 490-501.
34. Bingham, S.; Kot, M. Multidimensional trees, range searching, and a correlation dimension algorithm of reduced complexity. *Physics Letters A* **1989**, *140*, 327-330.
35. Bellman, R.; Bellman, R.E.; Bellman, R.E.; Bellman, R.E. *Adaptive control processes: a guided tour*; Princeton university press Princeton: 1961; Vol. 4.
36. Verleysen, M.; François, D. The curse of dimensionality in data mining and time series prediction. In *Computational Intelligence and Bioinspired Systems*, Springer: 2005; pp. 758-770.
37. Yianilos, P.N. Data structures and algorithms for nearest neighbor search in general metric spaces. In Proceedings of SODA; pp. 311-321.
38. Bozkaya, T.; Ozsoyoglu, M. Indexing large metric spaces for similarity search queries. *ACM Transactions on Database Systems (TODS)* **1999**, *24*, 361-404.
39. Uhlmann, J.K. Satisfying general proximity/similarity queries with metric trees. *Information processing letters* **1991**, *40*, 175-179.
40. Nair, A.S.; Sreenadhan, S.P. A coding measure scheme employing electron-ion interaction pseudopotential (EIIP). *Bioinformation* **2006**, *1*, 197.
41. Holden, T.; Subramaniam, R.; Sullivan, R.; Cheung, E.; Schneider, C.; Tremberger, G.; Flamholz, A.; Lieberman, D.; Cheung, T. ATCG nucleotide fluctuation of Deinococcus radiodurans radiation genes. In Proceedings of Instruments, Methods, and Missions for Astrobiology X; p. 669417.
42. Voss, R.F. Evolution of long-range fractal correlations and 1/f noise in DNA base sequences. *Physical review letters* **1992**, *68*, 3805.

43. Faloutsos, C.; Ranganathan, M.; Manolopoulos, Y. *Fast subsequence matching in time-series databases*; ACM: 1994; Vol. 23.
44. Mitsa, T. *Temporal data mining*; CRC Press: 2010.
45. Mörchen, F. Time series feature extraction for data mining using DWT and DFT. Univ.: 2003.
46. Jensen, A.; la Cour-Harbo, A. *Ripples in mathematics: the discrete wavelet transform*; springer: 2001.
47. Wu, Y.-L.; Agrawal, D.; El Abbadi, A. A comparison of DFT and DWT based similarity search in time-series databases. In Proceedings of Proceedings of the ninth international conference on Information and knowledge management; pp. 488-495.
48. Caboche, S.; Audebert, C.; Lemoine, Y.; Hot, D. Comparison of mapping algorithms used in high-throughput sequencing: application to Ion Torrent data. *BMC genomics* **2014**, *15*, 264.
49. Cotten, M.; Petrova, V.; Phan, M.V.; Rabaa, M.A.; Watson, S.J.; Ong, S.H.; Kellam, P.; Baker, S. Deep sequencing of norovirus genomes defines evolutionary patterns in an urban tropical setting. *Journal of virology* **2014**, *88*, 11056-11069.
50. Phan, M.V.; Anh, P.H.; Cuong, N.V.; Munnink, B.B.O.; van der Hoek, L.; My, P.T.; Tri, T.N.; Bryant, J.E.; Baker, S.; Thwaites, G. Unbiased whole-genome deep sequencing of human and porcine stool samples reveals circulation of multiple groups of rotaviruses and a putative zoonotic infection. *Virus evolution* **2016**, *2*.
51. Kiyuka, P.K.; Agoti, C.N.; Munywoki, P.K.; Njeru, R.; Bett, A.; Otieno, J.R.; Otieno, G.P.; Kamau, E.; Clark, T.G.; van der Hoek, L. Human coronavirus NL63 Molecular epidemiology and evolutionary patterns in rural coastal Kenya. *The Journal of infectious diseases* **2018**, *217*, 1728-1739.
52. Arias, A.; Watson, S.J.; Asogun, D.; Tobin, E.A.; Lu, J.; Phan, M.V.; Jah, U.; Wadoum, R.E.G.; Meredith, L.; Thorne, L. Rapid outbreak sequencing of Ebola virus in Sierra Leone identifies transmission chains linked to sporadic cases. *Virus Evolution* **2016**, *2*.
53. Agoti, C.N.; Otieno, J.R.; Munywoki, P.K.; Mwihuri, A.G.; Cane, P.A.; Nokes, D.J.; Kellam, P.; Cotten, M. Local evolutionary patterns of human respiratory syncytial virus derived from whole-genome sequencing. *Journal of virology* **2015**, *89*, 3444-3454.
54. Altschul, S.F.; Gish, W.; Miller, W.; Myers, E.W.; Lipman, D.J. Basic local alignment search tool. *Journal of molecular biology* **1990**, *215*, 403-410.
55. Menzel, P.; Ng, K.L.; Krogh, A. Fast and sensitive taxonomic classification for metagenomics with Kaiju. *Nature communications* **2016**, *7*, 11257.
56. Langmead, B.; Salzberg, S.L. Fast gapped-read alignment with Bowtie 2. *Nature Methods* **2012**, *9*, 357, doi:10.1038/nmeth.1923 https://www.nature.com/articles/nmeth.1923#supplementary-information.
57. Sović, I.; Šikić, M.; Wilm, A.; Fenlon, S.N.; Chen, S.; Nagarajan, N. Fast and sensitive mapping of nanopore sequencing reads with GraphMap. *Nature communications* **2016**, *7*, 11307.
58. Otto, C.; Stadler, P.F.; Hoffmann, S. Lacking alignments? The next-generation sequencing mapper segemehl revisited. *Bioinformatics* **2014**, *30*, 1837-1843.
59. Li, D.; Liu, C.-M.; Luo, R.; Sadakane, K.; Lam, T.-W. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics* **2015**, *31*, 1674-1676.
60. Anton, B.; Sergey, N.; Dmitry, A.; Alexey, A.; Mikhail, D.; Alexander, S.; Valery, M.; Sergey, I.; Son, P.; Andrey, D. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of Computational Biology* **2012**, *19*, 455.

654 61. Tapinos, A.; Mendes, P. A method for comparing multivariate time series with different dimensions.
655 *PloS one* **2013**, *8*, e54201.
656 62. Sheybani, E.O. An Algorithm for Real-Time Blind Image Quality Comparison and Assessment.
657 *International Journal of Electrical and Computer Engineering (IJECE)* **2011**, *2*, 120-129.
658 63. Hendriks, R.C.; Gerkmann, T.; Jensen, J. DFT-domain based single-microphone noise reduction for
659 speech enhancement: a survey of the state of the art. *Synthesis Lectures on Speech and Audio Processing*
660 **2013**, *9*, 1-80.
661 64. Shi, H.; Schmidt, B.; Liu, W.; Müller-Wittig, W. A Parallel Algorithm for Error Correction in
662 High-Throughput Short-Read Data on CUDA-Enabled Graphics Hardware. *Journal of Computational*
663 *Biology* **2010**, *17*, 603-615, doi:10.1089/cmb.2009.0062.
664 65. Zhang, Q.; Pell, J.; Canino-Koning, R.; Howe, A.C.; Brown, C.T. These Are Not the K-mers You Are
665 Looking For: Efficient Online K-mer Counting Using a Probabilistic Data Structure. *PLoS ONE* **2014**, *9*,
666 e101271, doi:10.1371/journal.pone.0101271.
667 66. Salikhov, K.; Sacomoto, G.; Kucherov, G. Using cascading Bloom filters to improve the memory usage
668 for de Brujin graphs. In Proceedings of WABI; pp. 364-376.
669 67. Berlin, K.; Koren, S.; Chin, C.-S.; Drake, J.P.; Landolin, J.M.; Phillippy, A.M. Assembling large genomes
670 with single-molecule sequencing and locality-sensitive hashing. *Nature biotechnology* **2015**, *33*, 623-630.
671 68. Camerra, A.; Palpanas, T.; Shieh, J.; Keogh, E. iSAX 2.0: Indexing and mining one billion time series. In
672 Proceedings of Data Mining (ICDM), 2010 IEEE 10th International Conference on; pp. 58-67.
673

674