Genome analysis

Influenza classification from short reads with VAPOR facilitates robust mapping pipelines and zoonotic strain detection for routine surveillance applications

Joel A. Southgate1,*, Matthew J. Bull1,2, Clare M. Brown1, Joanne Watkins2, Sally Corden2, Benjamin Southgate3, Catherine Moore2 and Thomas R. Connor1,2

1Organisms and Environment Division, School of Biosciences, Cardiff University, Cardiff CF10 3AX, UK, 2Public Health Wales, University Hospital of Wales, Cardiff CF14 4XW, UK and 3MRC Centre for Regenerative Medicine, University of Edinburgh, Edinburgh EH16 4UU, UK

*To whom correspondence should be addressed.

Received on June 4, 2019; revised on August 18, 2019; editorial decision on October 27, 2019; accepted on October 30, 2019

Abstract

Motivation: Influenza viruses represent a global public health burden due to annual epidemics and pandemic potential. Due to a rapidly evolving RNA genome, inter-species transmission, intra-host variation, and noise in short-read data, reads can be lost during mapping, and de novo assembly can be time consuming and result in misassembly. We assessed read loss during mapping and designed a graph-based classifier, VAPOR, for selecting mapping references, assembly validation and detection of strains of non-human origin.

Results: Standard human reference viruses were insufficient for mapping diverse influenza samples in simulation. VAPOR retrieved references for 257 real whole-genome sequencing samples with a mean of > 99.8% identity to assemblies, and increased the proportion of mapped reads by up to 13.3% compared to standard references. VAPOR has the potential to improve the robustness of bioinformatics pipelines for surveillance and could be adapted to other RNA viruses.

Availability and implementation: VAPOR is available at https://github.com/connor-lab/vapor.

Contact: southgateJA@cardiff.ac.uk

Supplementary information: Supplementary data are available at Bioinformatics online.

1 Introduction

Influenza viruses are enveloped, single-stranded, segmented negative-sense RNA viruses of the family Orthomyxoviridae. Influenza A and B have eight genome segments, with major antigenic recognition sites within the two proteins haemagglutinin (HA) and neuraminidase. Accumulation of point mutations within the antigenic recognition sites within the two proteins haemagglutinin (HA) and neuraminidase can result in host immune evasion, thereby causing annual seasonal epidemics (Petrova and Russell, 2018; Taubenberger and Kash, 2010). Current estimates suggest that seasonal influenza A and B cause 4–5 million severe infections (Tafalla et al., 2016) in humans with ~291,000–645,000 (Iuliano et al., 2018) deaths per year globally. Furthermore, influenza type A is a zoonotic virus infecting a wide range of avian and other non-human species (Bauut et al., 2018). These viruses have the capability to reassort leading to the emergence of new strains (Bouvier and Palese, 2008), which can result in pandemics.

Whole-genome sequencing (WGS) has been used to study the influenza virus genome for over a decade and is emerging as an important tool in research and surveillance (Holmes et al., 2005; McGinnis et al., 2016; Meinel et al., 2018; Rutvisuttinunt et al., 2013). Protocols have been developed (Zhou et al., 2009, 2014) that facilitate routine monitoring of isolates by public health organizations, as well as the study of transmission events (Houlihan et al., 2018; Meinel et al., 2018). Two important data sharing resources exist to this end; the NCBI Influenza Virus Resource (NIVR) (Bao et al., 2008), and the Global Initiative on Sharing All Influenza Data (Shu and McCauley, 2017), wherein over a hundred thousand influenza genome segment sequences can be found at the time of writing. Sequencing can be performed directly from clinical swabs with single-reaction genomic reverse transcription polymerase chain reaction (RT-PCR) (Goldstein et al., 2018; Meinel et al., 2018). Furthermore, bioinformatics pipelines have begun to be developed for efficient processing of this data (Borges et al., 2018; Wan et al., 2015).

Despite the increasing application of next-generation sequencing to influenza, the pitfalls associated with current bioinformatics approaches have not been explored in depth. Influenza virus de
**2.2 Mapping assessment**

Four mapping programmes were assessed in this analysis: Minimap2 (Li, 2018), BWA-MEM (Li and Durbin, 2009), NGM (Sedlazeck et al., 2013) and Hisat2 (Kim et al., 2015). These tools were used to represent a range of algorithms and intended use cases, in order to assess whether robustness to influenza data is a common problem, or whether it can be addressed by choice of tool. Default settings were used for all tools. Each experiment can be reproduced using the code and instructions found at github.com/connor-lab/vapor_mapping_benchmarking. Four mapping simulations were performed in total.

For assessment of the sufficiency of single reference strains for mapping diverse samples, two simulations were performed. For assessment of robustness to species origin, read sets were simulated with ArtificialFastqGenerator (Frampton and Houlston, 2012) from 552 avian, 16,679 human and 4054 swine H1N1 HA coding sequences from the NIVR (Bao et al., 2008). An additional 0.05% in silico substitution was introduced into simulated reads to account for RT-PCR technical errors and biological intra-host variation. This rate was chosen to be in accordance with experimental observations made by Orton et al. (2015). Reads were then mapped to the A/California/07/2009 (H1N1) HA reference sequence. For assessment of robustness to divergence, technical and biological noise, reads were simulated from A/Perth/16/09 (H3N2) HA, with additional in silico mutation with per-base rates between 2 and 16%, performed uniformly across the chosen reference sequence; reads were simulated as above, then mapped back to A/Perth/16/09 (H3N2). This was performed 1000 times for each mutation rate. A/California/07/2009 (H1N1) and A/Perth/16/2009 (H3N2) were used as references since they are common clade representatives. Samtools (Li et al., 2009) was used to retrieve successfully mapped reads, which were then counted.

For comparison of mapping with and without VAPOR classification, and potential zoonotic virus detection, two simulations were performed using 33,133 unique approximately full-length influenza A HA coding sequences of any lineage or species, downloaded from the NIVR. In the first case, 5000 pairs of sequences were chosen randomly; the first of the pair was used for read simulation as above, and the second as a mapping reference. In the second simulation, a single sequence was randomly chosen for read simulation, and the reference was chosen by VAPOR version 1.0.1. This process is shown in Supplementary Figure S1. As before, successfully mapped reads were extracted with samtools, then counted.

To assess the potential benefit of classification with VAPOR on real data, 206 of 257 fastq file pairs were subjected to mapping with VAPOR with default settings for short reads (<51 bp), both with and without VAPOR classification. A total of 51 out of 257 samples with <1000 HA reads were excluded to avoid very low coverage samples skewing calculation of mean percentage gain. In the first case, reads were mapped to a set of 4 HA references from different subtypes: A/Perth/16/2009 (H3N2), A/California/07/2009 (H1N1), B/Florida/4/2006 (Yamagata) and B/Brisbane/60/2008 (Victoria). In the second case, VAPOR was used to choose a single reference from 53,758 influenza A and B HA references. The number of reads mapping and the number passing VAPOR pre-filtering was recorded in each case.

**2.3 Algorithm overview**

**2.3.1 Definitions**

Let \( R = \{r_1, r_2, \ldots, r_k\} \) and \( S = \{s_1, s_2, \ldots, s_k\} \) be indexed multisets of strings (sequencing reads and references, respectively), over a common alphabet \( \Sigma = \{A, T, C, G\} \), where \( |R| \) denotes the cardinality of set \( R \). Let \( W = \{N, E, W\} \) be a weighted DBG built from reads \( R \), where \( N, E \) and \( W \) are sets of nodes (k-mers), edges (k−1-mers) and node weights (sequencing depth for some k-mer), for some \( k \geq 2 \) (by default \( k = 21 \)). We assume a model read generation process reflective of RNA virus sequencing: let the multiset \( X = \{x_1, x_2, \ldots, x_k\} \) be a population of virus sequences (quasispecies) for some gene, for which we suppose there is some major variant \( x^* \) with the greatest multiplicity. Let reads \( R \) be generated from this...
population, with varying coverage across the gene (possibly by several orders of magnitude), and additional errors (due to RT-PCR and sequencing). We attempt, using heuristics, to find a reference that is similar to $x$.

2.3.2 Mapping and scoring
VAPOR maps each reference $s$ against $W$, such that the $r$th $k$-mer of $s$, denoted $s[i..i+k]$ is either mapped to some node $n \in N$, or mapped to a gap. We note that $s[i..i+k]$ does not have to equal $n$. Let $s'$ be the string representation of the path mapped to by $s$.

Figure 1 demonstrates the concept of this mapping.

We next formulate a scoring function $f_{\psi}(s', s)$. We chose to favour sequences for which there is high weight in $W$; due to the high degree of variation in RNA virus datasets, and large number of closely related reference sequences, many $k$-mers may be present in the $W$ at low frequency, such that there may be several reference sequences which correspond exactly to a path in $W$. Conversely, since sequencing depth in these datasets may be highly skewed, we seek to also reward matches which cover a greater proportion of the reference, rather than those that have high depth for a short subsequence, then poor matches elsewhere. In order to capture this trade-off, we define:

$$ f_{\psi}(s', s) = \psi(s') \cdot \sum_{j=1}^{n} M_j \delta(s'_i, s_j) $$

(1)

where $\psi(s')$ is the fraction of non-gap bases of $s'$, $|s|$ is the length of string $s$ and $M_i$ is the maximum sequencing depth of $k$-mers that overlap with the $r$th base of $s$. That is, for the sequence of node weights $w_i$ with corresponding $k$-mer nodes $n_i$ of $s'$, $M_i = \max\{w_i^1, w_i^2, \ldots, w_i^k\}$, where $i = \max(0, i-k+1)$ and $\delta(s'_i, s_j) = 1$ if $s'_i = s_j$, and 0 otherwise. Since any reference can be mapped onto the graph in many ways, we attempt to heuristically find high scoring placements.

Fig. 1. Simplified VAPOR algorithm. Firstly, pre-processing and graph construction is performed (A), where target reads $R$ (solid black lines) are filtered from non-target (e.g. bacterial) reads (dotted lines) using a fast $k$-mer comparison to references $s$. This is followed by wDBG construction. Then, mapping and scoring is performed simultaneously (B), where each reference sequence $s$ (dashed line) is mapped to the wDBG, $W$, built from these reads. This is done in two main steps: exact $k$-mer matching (black circles) and extension (white circles) by heuristic graph traversal

2.3.3 Pre-processing
VAPOR first filters reads to remove non-target sequences (e.g. bacterial) and decide orientation of reads. As input, VAPOR takes a fasta file of full or approximately full-length reference segment sequences, and a.fasq (or.fastq.gz) file of WGS reads. Firstly, VAPOR builds a set of $k$-mers $U$ from all reference sequences. Next, the $r$th read is decomposed into a set of non-overlapping subsequences of length $k$ (words), $A$, and if $|A| \leq f$ (the proportion of read words also present in the references), where $|A|$ gives the number of elements of the set $A$, for some specified parameter $f$, the read is discarded. This is repeated for the reverse complement; if both are kept, the highest score decides orientation. Furthermore, in order to try to eliminate erroneous $k$-mers, any node $n_i \in N$ with corresponding weight $w_i \in W$ less than a coverage parameter $c$ is discarded.

2.3.4 Core algorithm
Firstly, $W$ is built from the filtered reads. Then for each input reference sequence, $s$, the core algorithm of VAPOR makes use of a heuristic seed-and-extend procedure to find a high scoring mapping of $s$ onto $W$. Each reference sequence, $s_i$, with length $|s_i|$, is decomposed into a sequence of $k$-mers. Querying proceeds in four phases, where the query is walked along the wDBG: $k$-mer seeding, trimming, bridging and scoring. We seek to simultaneously perform the mapping and compute the array $M' = (M_1\delta(s_1, s_1'), M_2\delta(s_2, s_1'), \ldots, M_{|s|}\delta(s_{|s|}, s_1'))$ as in (1). Firstly, an array $a$ is initialized from exact $k$-mer matches, where $a_{n_m}$ is the weight of the $m$th $k$-mer of the reference, and any not in $N$ are set to zero. For speed considerations, only a subset of seed arrays is extended: those with a fraction of non-zero elements greater than a user-defined parameter --min_kmer_cov (default: 0.1), and in a top user-defined percentile --top_seed_frac (default: 0.2). In order to reduce the number of suboptimal exact matches, seeds are trimmed. Each seed (sequence of $k$-mer matches) in the array $a$ is trimmed back (set to zero) at both ends until a suboptimal branch points in the graph within $\rho$ positions of the end of the seed is found. This procedure is used to heuristically prevent suboptimal seeds to low coverage regions of the wDBG, possibly generated by error or low frequency variants. Next, bridging is performed. For the $r$th gap (run of zeros) in $a$ of length $l$, a bridge $b$ is formed by walking $l$ locally optimal (where there is a branch, the edge with the highest weight) edges in the wDBG from the last matching $k$-mer. As such, bridging attempts to extend a mapping with only exact matches to one with inexact matches. Next, the array $M$ is computed by (i) inserting bridge $k$-mer weights and (ii) re-calculating the weight at each position $j$ as $M_j$ (as defined in mapping and scoring). Finally, each bridge, $b$, a string, is then compared to the $r$th gap string, the original substring in the reference sequence corresponding to the gap, in order to compute $\delta(s'_i, s_j)$ as in (1). For any $s_j$ in an exact match, $\delta(s'_i, s_j) = 1$ by definition. Figure 2 shows the steps involved in computing the array $M$ for an example graph mapping.

VAPOR is implemented in Python3, with source code available at github.com/connor-lab/vapor.

2.4 Classification benchmarking
VAPOR was compared to MASH (Ondov et al., 2016) and BLAST (Altschul et al., 1990) read classification by simulation. BLAST consensus classification was performed by BLASTing each read, taking the best scoring references by $e$-value then bit-score, summing the number of times each result occurs in all reads and returning the most frequent. Reads were simulated as follows: a reference, $s_0$, was chosen from 46,724 uniquely approximately full-length influenza A HA sequences from the NIVR, and mutated uniformly with a given probability (0.01, 0.02, 0.03) to generate a mutated sequence $s_m$; reads were simulated with ArtificialFastqGenerator as before, with a higher uniform error rate of 1%, in order to provide a challenging classification task representative of difficult datasets. To provide an additional challenge, we simulated an intra-host population with four minor sequences, mixed in the ratio of 100:5:1:1, with each
2.6 Computational resources and performance benchmarks

In all cases, experiments were performed natively on a 96 core, 1.4 TB memory CentOS version 7.4.1708 virtual machine hosted by Cloud Infrastructure for Microbial Bioinformatics (CLIMB) (Conneor et al., 2016), with GNU parallel (Tange, 2011) where required. Basic time and space benchmarks comparing BLAST and VAPOR were performed with GNU-time (wall-clock time, maximum resident set size). The largest 11 influenza A samples with at least 1% (between 1 and 16%) of reads identifiable as influenza (samples 36, 51, 93, 95, 100, 101, 105, 113, 114, 131, 153) were randomly sub-sampled to between 200 000 and 2 000 000 reads (total for a pair of fastq files). BLAST and VAPOR were used to query a database of 47 073 influenza A HA gene sequences. For BLAST, reads were first converted to fasta files, which was not included in the benchmark time. For BLAST, tabulated output was specified (-outfmt 6), and output redirected to/dev/null in order to reduce I/O time (assuming a tool using BLAST for classification would not parse output files). VAPOR was also benchmarked on a laptop with an Intel® Core™ i7-6600U CPU @ 2.60 GHz with 8 GB of memory, using the same samples without sub-sampling (mean read total 6 600 000).

3 Results

3.1 Benchmarking single-reference mapping

A range of mapping programmes (Minimap2, BWA-MEM, Hisat2 and NGM) were compared to assess possible data loss when single references are chosen for mapping of short reads from influenza virus WGS datasets. For the first experiment, simulated reads from 16 679 human, 552 avian and 4054 swine H1N1 HA sequences retrieved from the NIHVR were mapped to the reference strain A/California/07/2009 (H1N1). Reads were simulated with an additional 0.05% error on top of simulated sequencing error to account for the combined effect of intra-host population variation and RT-PCR error. This error rate was found to be conservative when compared to the raw error rate in our datasets, as shown by Supplementary Figure S5, which was frequently higher than 2%. The proportion of successfully mapped reads for each tool and host species is given in Figure 3. In this case, using a single reference strain with any of the programmes resulted in unmapped reads. NGM resulted in the lowest average percentage of unmapped reads. When utilizing a database of all H1N1 sequences from human hosts, Minimap2, NGM, BWA-MEM and Hisat2 had mean mapping percentages of 87.2, 92.2, 89.1 and 84.9% respectively; as such, even for these influenza sequences, data loss was not uncommon, possibly due to samples in the database representing human infection from zoonotic strains. However, for avian and swine samples, read recovery was poor. For NGM, only 34.1% of avian reads mapped successfully on average. Swine sequences were mapped with intermediate success. This provides evidence that, should zoonotic strains be sequenced in routine surveillance, they may fail to map entirely, and go undetected.

Secondly, in order to assess how read recovery varies with sequence divergence, reads were simulated by taking the coding sequence of A/Perth/16/09 HA and subjecting it to in silico uniform mutation at specified rate, with additional read error of 0.05% as before (Supplementary Fig. S4). For all mapping programmes, at ~10% mutation, read recovery begins to regularly diminish, which is insufficient for robust mapping of influenza strains from different species. Furthermore, for several of the programmes tested, mapping quality was suboptimal beyond 1–3% mutation.

3.2 Simulation classification performance

In order to assess the performance of classification from simulated reads, our tool, VAPOR, was compared to MASH and consensus BLAST classification. Reads were simulated from randomly selected
H1N1 HA sequences mutated with a given uniform per-base probability, with additional read error of 1% to provide challenging datasets. A fourth category included simple simulated intra-host populations (denoted as 3%/Q). Figure 4 shows the additional Levenshtein distance, \( \Delta L \), for each tool. Mean coverage for simulated reads was 77.76 for single-sequence simulations, and 96.03 for simulated intra-host populations. The average additional distance of retrieved sequences for MASH were 4.69, 5.24, 6.83 and 7.28, showing some sensitivity to additional simulated variant noise; for all cases mean additional distance for BLAST and VAPOR were below 0.74 and 0.88, respectively. For MASH, the 75, 95 and 99% percentiles for retrievals for the 3% threshold were 11.00, 24.00 and 37.04. However, for BLAST and VAPOR, these percentiles were under 12 and 14, respectively, for all cases. These results show that references chosen by BLAST and VAPOR were often near-optimal or optimal, despite a large amount of noise, and that the performance difference between these approaches was small. These results show that the algorithm used by VAPOR facilitates accurate classification of influenza strains directly from reads, comparable in accuracy to BLAST for WGS read sets, which is generally not computationally tractable for datasets with millions of reads (shown in Supplementary Fig. S5).

### 3.3 Real data classification performance

In order to validate the performance of VAPOR directly on real datasets, we took raw reads from 257 samples corresponding to 1495 segment contigs previously processed and assembled with IVA, with a single full-length contig each previously annotated by BLAST. In each case, corresponding reads were classified by VAPOR. The chosen reference was compared by global alignment to the assembled contig versus the PID of references selected by BLAST classifications of contigs. Comparison to BLAST classification of contigs was used to provide a baseline near-optimal classification. The mean PID between contig and VAPOR classification was 99.82%. In the case of NS1, VAPOR outperformed BLAST annotation of assembled contigs, with a mean of 99.48 versus 98.74%. On closer inspection, this was a result of the method used to sort BLAST results combined with the presence of a sequence with an additional 150–200 bp of the 3’ untranslated region (UTR). Removal of this sequence and sorting by PID resulted in a mean of 99.45 and 99.93%, respectively. These results show that in most cases tested, VAPOR was able to accurately identify a sample from reads with comparable performance to BLAST annotation of assembled contigs. We note that, for some contigs, neither BLAST nor VAPOR could achieve classifications with a PID >97%. Manual examination of these samples showed large deletions, with at least one a likely misassembly (deletion including start codon, inclusion of 5’ UTR).

### 3.4 Mapping with pre-classification

Raw mapping performance was also assessed on real data by mapping datasets with Minimap2 with and without pre-classification. Figure 6 shows the number of additional reads mapped when pre-classification was performed. In all but one case, this resulted in a greater number of mapped reads, with a mean of 7816.03, corresponding to a mean percentage gain of 6.83%, including a case with over 68 000 additional reads. The maximum percentage increase was 13.32%. An outlier did occur where the number of mapped reads decreased. In this case, VAPOR identified several thousand more reads as influenza than were mapped. On further inspection, for this sample, reads mapped to both A/Perth/16/09 (H3N2) and A/California/07/09 (H1N1), indicating that the sample represented influenza from two different subtypes. As such, this sample represented a true biological coinfection or a contamination and could not be mapped to a single reference. For simulations with randomly selected pairs from any species, mean recovery rates were <20% without pre-classification (Supplementary Fig. S5). However, with pre-classification using VAPOR, the mean was over 99.72% for all tools, demonstrating...
false positive rate for $v$ of 91.35%. This is expected because, as previously shown, VAPOR generally was able to classify strains to within a few base-pairs; randomly chosen zoonotic strains generally had PIDs of <90% to human strains, depending on origin. We note that, given the database used, some avian strains may have been isolated from humans, and labelled as human; as such, perfect classification with this dataset may be impossible. In order to provide a more difficult reassortment detection task, the same experiment was performed between human H3N2 sequences. We found at a PID threshold of 96.3%, a true positive rate of 76.8% could be achieved at a false positive rate of 10.8%. This result was expected given that sequences from different H3N2 strains generally have a PID within a few percent. In total, these results provide evidence that reassortments with zoonotic strains can be detected directly from reads with reasonable accuracy, but that intra-lineage reassortments may be more difficult.

3.6 Run-time and computational requirements

For benchmarking performed on the CLIMB VM, VAPOR performed all benchmarks within 11 min. As expected, using BLAST to query all reads was too slow for general application to these datasets, as shown in Supplementary Figure S7. For the 2 000 000 read samples, BLAST took over 20 h in the worst case, excluding both pre- and post-processing of results. For the laptop computer, VAPOR classified samples with a mean of 6 600 000 total reads in a mean time of 3.73 min and mean peak memory consumption of 2.78 Gb.

4 Discussion

4.1 Mapping approaches and improvement with VAPOR

We provide evidence that, in the best case, approaches for influenza virus analysis that use mapping to a single reference may result in data loss due to biological variation and noise. As shown in Supplementary Figures S2 and S3, influenza strains continually accumulate substitutions relative to a single reference (~5 substitutions per year for H3N2) and reads may have a high error rate (>2%). Often, mapping to a single reference may be most unreliable for important samples, such as zoonotic transmission events. In those worst cases, mapping may fail completely, when usable data are present, requiring time and expertise to resolve with more complex methods. Our approach largely avoids these pitfalls altogether, allowing simpler pipelines and retaining the advantages of using a mapping-based approach for analysis. We chose Minimap2, BWA-MEM, NGM and Hisat2 in order to represent a range of mapping softwares. BWA has found use in general for influenza read mapping (Borges et al., 2018; Imai et al., 2018; Jonges et al., 2014; Leonard et al., 2016; Rutvisuttinunt et al., 2013; Wu et al., 2014; Yu et al., 2014). In some cases, references were chosen by mapping-based approaches for selection (Yu et al., 2014). Of these softwares, only NGM was developed with specific robustness to variation. Furthermore, the experiments reported were not intended as complete evaluations of the programmes, since such an evaluation must also include mapping quality. Our data, however, does demonstrate that pre-classification with raw reads provides a broad strategy to improve robustness of pipelines and achieve faster results. For the chosen references, A/Perth/16/2009 (H3N2) and California/07/2009 (H1N1) were chosen as vaccine strains recommended by the WHO multiple times, and have also been used previously as references (Rutvisuttinunt et al., 2013; Simon et al., 2019). In other cases, different single references have been used (Meinel et al., 2018). We do not believe that using different individual strains would affect the trends demonstrated.

We note that alternative approaches exist, including mapping to a large sequence database, but this does not easily allow for visualization of an alignment to a single reference, and subsequent analysis such as characterization of point mutations. We note that in principle, pre-classification with any software could work reasonably well. MASH performed well in simulations. However, using an
Acknowledgements

We would like to acknowledge and offer our thanks to the Pathogen Genomics Unit (PenGU) at Public Health Wales for their efforts and assistance in performing laboratory work, without which this research could not have been performed. Likewise, we would also like to thank Angela Marchbank and the Sequencing Hub at Cardiff University for the sequencing work performed.

Funding

J.S. was supported by the Biotechnology and Biological Sciences Research Council-funded South West Biosciences Doctoral Training Partnership [training grant reference BB/M009122/1]. This work made use of computational resources provided by Medical Research Council Cloud Infrastructure for Microbial Bioinformatics, which also funds T.R.C. and M.J.B. [grant reference MR/L015080/1]. The project was also supported by specific funding from Welsh Government which provided funds for the sequencing of influenza isolates. The Pathogen Genomics Unit (S.C., J.W.) was funded through Genomics Partnership Wales by Welsh Government.

Conflict of Interest: none declared.

References


Influenza classification from short reads with VAPOR

optimal reference is ideal, since for later advanced applications, such as transmission events, or study of intra-host variation, the closest possible reference may be necessary. Furthermore, VAPOR permits simultaneous filtering out of any non-human or bacterial reads with optimal reference selection. Whilst BLAST performed well for individual read classification, it is often too slow for general application. Furthermore, assembly of virus genomes can be slow, often taking several days for a single sample when contaminant reads, such as human DNA, are present. Finally, misassembly can occur (Wymant et al., 2018), which potentially occurred in several of our samples.

In all but one of our real-data cases examined, pre-classification with VAPOR resulted in a greater number of mapped reads than mapping to four reference strains A/H3N2, A/H1N1, B/ Victoria and B/Yamagata. However, for a single sample, which contained influenza sequences from two clades, the number of mapped reads was reduced. Although VAPOR can report the number of influenza sequences detected in total, future study should be utilized to develop methods of co-infection detection. In these relatively rare cases, a single reference is not sufficient for mapping.

4.2 VAPOR algorithm and performance

We have presented a novel approach to virus classification from short-reads data using DBGs. In future study, as public sequence data accumulates, our algorithm may show promise in WGS approaches for other RNA viruses with small genomes, such as measles virus, human immunodeficiency virus or Ebola virus. Several default parameters were explored during development, but not exhaustively. A 1-kmer size of 21 was utilized, as this was also able to perform read pre-filtering from contaminating sequences, without addition of a separate parameter. Similarly, parameters controlling the minimum fraction of required k-mers for seed extension, as well as the top percentile of seeds chosen for extension could be adjusted, possibly to improve speed. However, in the read sets examined, the default parameters were generally sufficient to ensure matches were found and did not appear to exclude potentially optimal matches. However, for novel strains that differ greatly from all strains previously observed, more sensitive parameterizations may be required.

4.3 Real-data classification

As shown in Figure 4, we note that the BLAST contig classification strategy we used performed poorly on NS1. This was due to sorting by e-value, bit-score and length over PID, combined with the presence of some NS1 sequences in the database which were longer than the required coding region. Since in general usage we do not wish to exclude sequences with longer 3′ UTRs, we opted to include this result to illustrate a potential pitfall that can occur with automated BLAST classification, as well as the trade-off between length and PID. Although sorting by PID may alleviate this problem, it may also yield shorter, incomplete alignments. For some samples, neither BLAST nor VAPOR could retrieve a sequence closer than 96% to the assembled contig. For some samples, this was due to large deletions present in the assembled contig. Although some of these deletions may be present in the true biological sequences, for at least one, this was due to suspected misassembly. These assemblies were also included to draw attention to potential problems that may be encountered during analysis.

5 Conclusion

Here we demonstrate that influenza sequence pre-classification with VAPOR minimizes data loss, reduces pipeline complexity, and allows for classification of zoonotic strains and reassortments directly from reads. We believe that the simplicity of our approach has potential to alleviate several difficulties associated with current bioinformatics pipelines and could reduce workloads in public health surveillance. Lastly, whilst we have tested VAPOR extensively for use with influenza, we believe our approach may be more broadly applicable to other sequence data, particularly small RNA and DNA viruses.