DEVELOPING BIOINFORMATICS APPROACHES FOR THE ANALYSIS OF INFLUENZA VIRUS WHOLE GENOME SEQUENCE DATA

Joel Alexander Southgate

A thesis submitted to Cardiff University for the degree of

Doctor of Philosophy

March 2021

Acknowledgements

I would like to thank my supervisors Thomas Connor and Andrew Sewell. Thanks are extended to Thomas Connor for his guidance and advice over the course of the project. I would also like to thank him for encouraging independent thought and the freedom to pursue my academic interests.

I would like to thank Matthew Bull for all the support and expertise he has offered over the years, which cannot be overstated.

I would like to thank my assessor Peter Kille for his advice and insight over the years.

I would like to thank my DTP friends and peers, Sion Edwards, Jordan Cuff, and Edward Cunningham-Oakes, for their support and encouragement.

I extend thanks to the BBSRC for funding my work, and the members of the SWBio team for all of their efforts.

I would like to thank the members of the CLIMB project, who provided computational resources.

I would additionally like to thank the members of Public Health Wales for their hard work and collaboration.

For their love and support, I would like to thank Joanna Redihough, Janice Morgan, David Southgate, Ben Southgate, and Imogen Southgate. I would like to thank Beatrice Morgan for her constant positivity and belief in me. Additionally, I would like to thank the extended members of the Morgan and Southgate families for their help over the years.

Dedication

This thesis is dedicated to my best friend and fiancée Joanna Redihough.

Summary

Background

With the rise of virus genomics, we are now, more than ever, accumulating virus sequence data at an astonishing rate. Now, online databases feature hundreds of thousands of genome sequences from viruses such as influenza, HIV, or SARS-CoV-2. Sequencing in real-time has become possible. In the face of this unprecedented scale, the need for high-performance bioinformatics methods for virus whole genome sequencing pipelines has never been so great.

Methods

Broadly, two components of the RNA virus whole genome sequencing pipeline were approached, with a particular focus on automation. In the first part, several methods for optimizing the sequence reconstruction process were developed using several hundred influenza whole genome sequencing samples from two seasons, including: a graph-based algorithm for reference selection; detection of contamination and coinfection using mixture modelling; and fast virus genome comparison. In the second part, phylogenetic analyses were developed and tested, making use of both the influenza dataset and a collection of thousands of SARS-CoV-2 genomes. These analyses included: benchmarking of methods for molecular dating of influenza virus data; phylodynamic exponential growth modelling for SARS-CoV-2; and application of phylogenetic methods for inferring importation for both influenza and SARS-CoV-2.

Results

A full whole-genome sequencing pipeline for influenza was implemented. Within this, multiple components were developed and integrated with existing tools. Firstly, a graph-based algorithm was successfully implemented for reference selection, whereby reference-based mapping was identified as a source of potential bias. In addition to this, mixture modelling was demonstrated to be applicable to the task of detecting potential contamination post-assembly. Next, a variant of the diagonal edit distance algorithm was developed to allow rapid exhaustive nearest-neighbor search for virus whole-genome sequences, although for hundreds of thousands of sequences a seed-and-extend approach is expected to be superior. In section two, phylogenetic and phylodynamic methods and analyses were assessed for use in small geographical regions, on short time-scales, with focus on single epidemics within Wales. I found methods for molecular dating to have varying performance in this context, and make recommendations for their application. Lastly, it was found that WGS data and molecular dating could be used in practice for routine surveillance, in particular to assess signatures of importation and geographical spread of influenza and SARS-CoV-2.

Contents

Chapter 1

Introduction

1.1 Influenza viruses

Influenza viruses are enveloped, single-stranded negative-sense RNA viruses of the family Orthomyxoviridae. Influenza types A and B represents a major public health burden worldwide (Woolthuis et al.; 2017)(Cox and Subbarao; 2000). These viruses cause an estimated 4-5 million severe infections, with up to 500,000 deaths per year (Tafalla et al.; 2016). Influenza A in particular has demonstrated potential to cause devastating pandemics; in 1918 'Spanish flu' resulted in the deaths of approximately 50 million individuals (Morens et al.; 2010). As such, influenza viruses represent pathogens of major importance. Considerable effort has been expended by international public health initiatives such as the World Health Organization (WHO) Global Influenza Surveillance and Response System (GISRS). To this end, whole genome sequencing (WGS) has been used to study influenza virus populations for over a decade, and has emerged as an important tool in research and public health surveillance (Holmes et al.; 2005a)(McGinnis et al.; 2016a)(Rutvisuttinunt et al.; 2013)(Meinel et al.; 2018a). Protocols have been developed (Zhou et al.; 2014)(Zhou et al.; 2009) that facilitate routine monitoring of isolates by public health organizations, as well as the study of transmission events (Meinel et al.; 2018a)(Houlihan et al.; 2018b). Two important data sharing resources exist to this end; the NCBI Influenza Resource (NIR) (Bao et al.; 2008a), and the Global Initiative on Sharing All Influenza Data (GISAID) (Shu and McCauley; 2017), wherein over a hundred thousand influenza genome segment sequences can be found at the time of writing, from isolates sampled across the globe. Methodologies exist for sequencing directly from clinical swabs with single-reaction genomic RT-PCR (Goldstein et al.; 2017)(Zhou et al.; 2009). Furthermore, bioinformatics pipelines have begun to be developed for efficient processing of this data (Borges et al.; 2018)(Wan et al.; 2015).

Influenza A and B viruses possess a genome of 8 RNA segments (Bouvier and Palese; 2008), each of which encodes for one of HA, NA, Polymerase basic 1 (PB1), PB2, Polymerase acidic (PA), Nucleoprotein (NP), non-structural 1 (NS1), and matrix protein 1 (M1), as well as several other smaller proteins, distinct in each type. Two fundamental evolutionary processes shape the influenza genome: mutation and reassortment. HA and NA in particular are subject to immunological selection, since these are the major antibody targets. This process is termed *antigenic drift*, which is the primary mechanism whereby strains avoid natural or vaccine-induced immunity (Taubenberger and Kash; 2010). Novel antigenic variants of A/H3N2 appear every 3-5 years, and 3-8 years for IVBs and A/H1N1 (Petrova and Russell; 2018).

1.2 The anatomy of a virus whole genome sequencing pipeline

Many examples of virus genomics pipelines exist in the literature and beyond, including: INSaFLU (Borges et al.; 2018); MAJORA (Nicholls et al.; 2020); ViraPipe (Maarala et al.; 2018) for general virus metagenomics; V-pipe (Cespedes et al.; 2020) for general; viral-ngs, provided by the Broad institute (unpublished but widely used, see `https://viral-ngs.readthedocs.io/en/latest/`); VirusDetect (Zheng et al.; 2017) for virus discovery from small RNAs; and many others (Chen, Huang and Sun; 2019; Bhuvaneshwar et al.; 2018; Li et al.; 2016). Typically, different laboratory protocols, experimental designs, hosts and viruses, lead to different bioinformatics requirements, and as such, custom pipelines are often developed to address specific challenges, although some authors have attempted to make large, general purpose pipelines. However, a few general tasks are performed by most pipelines, although the implementation details may vary. Figure 1.1 summarizes a few of these components of a virus sequencing pipeline used in routine epidemiology, from a software perspective.

1.3 String algorithms and their applications in RNA virus bioinformatics

In the process of whole genome sequencing, complete genomes are not read directly from the sequencer. Instead, fragments called reads are generated, often in extremely large quantities. In microbiological research, these fragments are assembled into genomes, which in turn are the subject of many different types of search queries. However, different organisms present different challenges. For RNA viruses, complex intra-host population structures mean that, instead of a single genome, many thousands of viral genomes are mixed together and then sequenced. This can present chal-

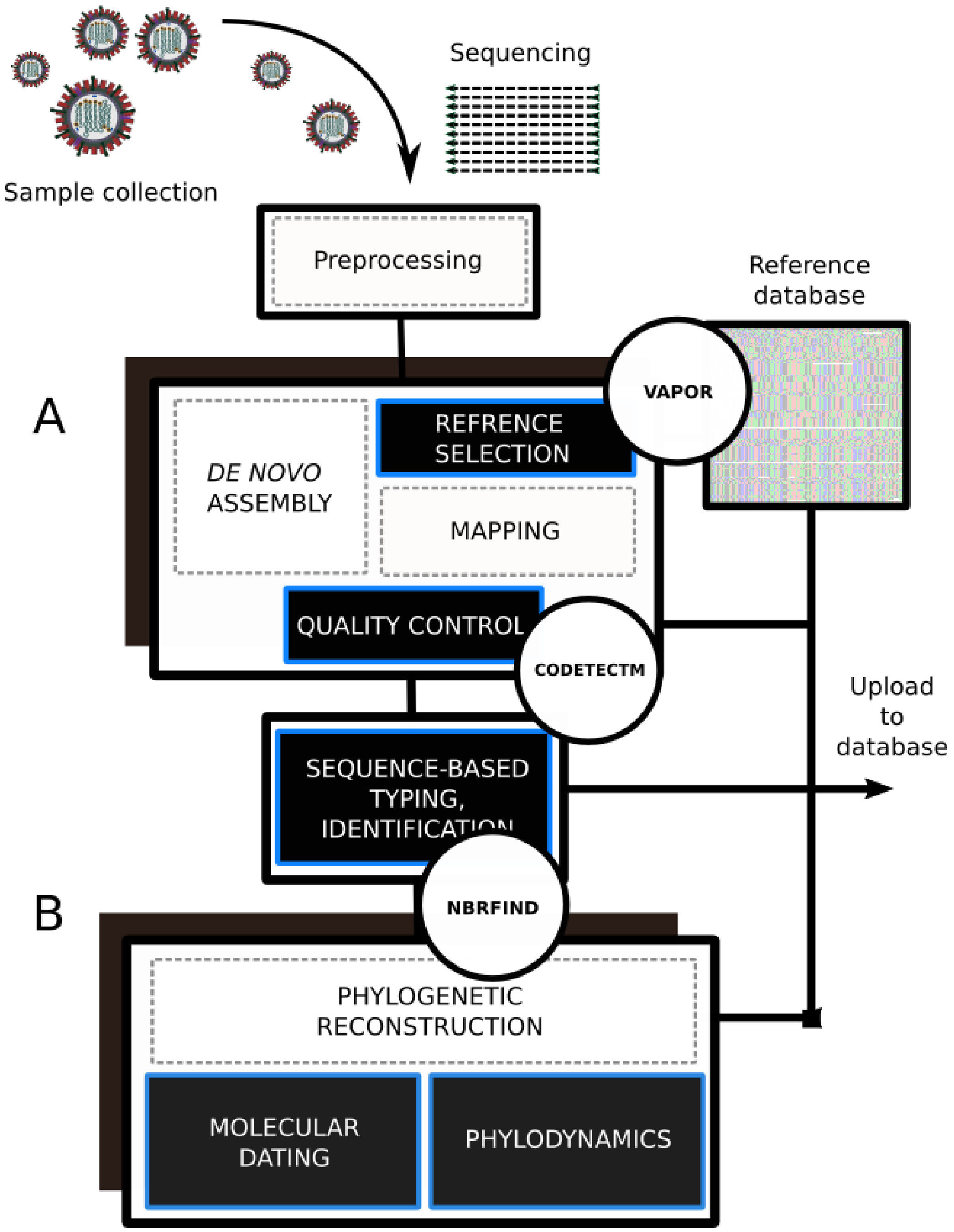


Figure 1.1: **Simplified schematic of an influenza virus sequencing pipeline**. After sample collection and sequencing, reads are preprocessed and subsequently fed into the pipeline. Firstly, reads are assembled, whether that is *de novo* or reference-guided, and resultant assemblies are filtered for quality (A). Then, analyses making use of groups of finished sequences are performed (B). These analyses generally involve phylogenetic tree building, molecular dating, and phylodynamics. These activities may require appropriate subsetting of data into relevant groups. Indicated in black are activities that are explored during the course of this thesis. Specifically developed software components are represented by circles.

lenges. Similarly, many RNA viruses are the subject of massive sequencing initiatives, which, in combination with a high mutation rate, can mean that typical search strategies may be suboptimal. As a result, software designed specifically for virus data in mind has emerged, building on previous methods. Here, I provide a brief review of important work underpinning virus bioinformatics, including: a) fundamental definitions, data structures and algorithms in string matching, often also used in work outside of bioinformatics; b) data structures and algorithms used in bioinformatics search algorithms, such as string indexing; c) specific applications to *de novo* assembly and classification of virus sequence data. Although these subjects span several fields within computer science and bioinformatics, often within the functionality of a single piece of software, many different methods must be drawn upon. The primary aim of this review is to provide a foundation for development of software and contextual discussion in subsequent chapters.

1.3.1 Fundamental string matching

Here, some fundamental definitions and basic algorithms for string matching are briefly described. Although the basic algorithms in particular are not necessarily widely used in bioinformatics, they constitute an importation foundation.

Strings

A string is a finite sequence S of symbols over an alphabet set Σ, where each $c \in \Sigma$ is termed a symbol (Sipser; 2012). For example, AAT is a string over the alphabet $\{A, C, G, T\}$. Let ϵ denote the empty string. Furthermore, let Σ^* denote the set of all strings of finite, non-zero length, that is $\Sigma^* = \bigcup_{k \in \mathbb{N}} \Sigma^k$. A substring of S is a contiguous sequence of characters of S, which is denoted $S[i : j]$ for indices i and j (left inclusive, right exclusive, as in the Python programming language). For some string S, a k-mer of s is a substring of length k (Vyverman et al.; 2012). For example, AAT is a k-mer of $AATTT$ of length 3. k-mers are widely used within bioinformatics (Melsted and Pritchard; 2011).

Graphs

A directed graph is the tuple $(\mathcal{N}, E)$, composed of a set of nodes $\mathcal{N}$, and edges E that connect them (Sipser; 2012). Edges are described as pairs (v, u) for nodes v and u in $\mathcal{N}$. For an undirected graph, (v, u) and (u, v) are equivalent.

Finite Automata

Finite automata have found use in bioinformatics, such as in: motif searching and discovery (Marschall and Rahmann; 2009; Marschall; 2011); protein classification (Psomopoulos et al.; 2004); phylogenetics (Westesson et al.; 2011; Holmes; 2017; Westesson et al.; 2012); and systems biology (Yang et al.; 2010; Sütterlin et al.; 2009). Since some relevant algorithms are formulated as finite state machines, we briefly summarize the definition of finite automata. Formally, a *finite automaton* (FA) M is defined (Sipser; 2012) by the tuple $(Q, \Sigma, \delta, q_0, F)$, where Q is a set of states (which the automata may take), Σ the set of characters comprising the alphabet of the input, $\delta : Q \times \Sigma \to Q$ the transition function (which determines how a given state transitions to a new state when combined with input), q_0 the initial state, and the set of acceptance states $F \subseteq Q$ (which are comparable to those states for which a boolean value of 1 is returned). For any string that is inputted into M, the final state after the last input is either in F or not, which means we can define a notion of accepted strings. For a set of accepted strings A, we define $L(M) = A$, and say that A is the language of M.

We extend this definition to the nondeterministic finite automata (NFA). In this case, the transition function is $\Delta : Q \times \Sigma \to \mathcal{P}(Q)$, where $\mathcal{P}(Q)$ is the power set of Q. In practical terms, this means that for any state and input, the transition may be made to more than one other state, and also that transitions can also occur without an input character (termed ϵ-transitions). FA without these features are described as *deterministic finite automata* (DFA). All NFA are equivalent to DFA; that is, they can be transformed by consideration of a DFA with states that are themselves combinations of $2^{|Q|}$ finite states (Marschall; 2011). Although this may be costly, there are algorithms for construction of a minimal DFA (that is one that accepts the same language with a minimal number of states) (Hopcroft; 1971).

Basic exact substring matching algorithms

One of the most important exact substring matching algorithms is the The Knuth-Morris-Pratt (KMP) algorithm. KMP makes use of the self-similarity of a word W in order to perform matching to some text S, reducing the time complexity of exact matches from $O(|W||S|)$ to $O(|W| + |S|)$ (Morris and Pratt; 1970; Knuth et al.; 1977). The algorithm allows us to ‘skip’ along the text, which is conceptually similar to the Boyer-Moore algorithm (Boyer and Moore; 1977; Tarhio and Ukkonen; 1993) (though the algorithms are different). For example, when scanning $W = AACAAT$ to $S = AACAACAAT$, we may encounter the mismatch at the end of $AACAAT$ and $S[0 : |W|] = AACAAC$. Naively, we can increment our position in S to start next at 1, and start checking $AACAAT$ against $S[1 : |W| + 1] = ACAACA$. However, we know already that $W[: -1] = AACAA$ matched $S[0 : |W| - 1]$. Since this is true, it cannot be that $W[: -1] = AACAA$ matches $S[1 : |W|]$, because if this were true, $AACA$ would match itself, but shifted over by one position, which we know to be false. Since we know this to be false, we can instead move to the next position for which this could be true, which coincides with $S[3 :] = AACAAT$. We can perform this with a useful array L, which indexes the correct position.

In this case, it is $L = [0, 1, 0, 1, 2, 0]$. How do we construct this? We iterate over prefixes $W[: i]$; for each new, encountered base w_i, we ask if it matches another base w_j, indexed by j. If so, we set $L_i = L_{i-1} + 1$ and increment i and j. If not, we set $L_i = 0$ and $j = 0$, and increment i. The KMP algorithm can be formulated as a DFA. The Aho-Corasick (AC) algorithm is similar to the KMP algorithm, except that we use a dictionary of words (Aho and Corasick; 1975). Using the KMP for each word, one at a time, would clearly result in additional complexity proportional to the number of dictionary words. The AC algorithm avoids this by construction of a prefix trie, augmented with suffix links. This is often formulated as a DFA. When a mismatch occurs, suffix links are followed and the transition is made at a different position. Other algorithms exist, such as the Rabin-Karp algorithm (Karp and Rabin; 1987), which makes use of a 'rolling hash function', and importantly forms a precursor to minimizers (Schleimer et al.; 2003).

1.3.2 Indexes and data structures for biological sequences

Although work summarized in the previous subsection was historically crucial, algorithms such as the KMP are often unsuitable for biological sequences, and alternative methods, especially based on indexes, have been developed. An index, in common usage, refers to something that helps us to locate something else more easily. Examples of these are keys in associative arrays. Within bioinformatics, a search index is often some form of data structure that allows us to find strings more easily. An example of this, as will be described, is the suffix tree, which allows substrings to be queried in linear time (Gusfield; 1997), which is an example of a substring index (Grossi and Vitter; 2005). Efficiently indexing collections of strings is crucial in many tasks within bioinformatics (Vyverman et al.; 2012), such as mapping (Li and Durbin; 2009d). The indexed string-matching problem differs from the string matching problem in that the text or collection of reference strings is preprocessed into an index which can then be used to accelerate the search (Crochemore et al.; 2014). As we shall see, in some cases, often a useful search index can also serve as a compressed representation of the underlying sequence data. Furthermore, the applicability of a given structure depends on the features of the data and the possible trade-offs between memory and processing requirements. Finally, we make the distinction between k-mer indexes (including BLAST) and full-text indexes (Vyverman et al.; 2012).

Suffix trees

Suffix trees are important full-text substring indexes that allow fast (and approximate) substring search (Weiner; 1973; McCreight; 1976). They can be considered compressed versions of the suffix trie, and are considered to be of key importance in the field of string processing (Abouelhoda et al.; 2004). Suffix tree construction can be performed in both $O(n)$ time and space (Ukkonen; 1995; Abouelhoda et al.; 2004). The suffix tree is also equivalent to a DFA known as a deterministic acylcic word graph (Navarro; 2001), which is similar to a sufffix trie, except that it is not a *tree*, and hence has $O(|S|)$ nodes (some states may join back up with another state later). The language accepted by a suffix tree is that of every substring of S. Similarly, the suffix automaton is a DFA that accepts the language given by suffixes of S. Despite linear space suffix tree construction algorithms, space requirements can still be large in practice, and data locality can be poor (Abouelhoda et al.; 2004). The feasibility of practical application of the suffix tree may depend on the precise problem. Basic problems such as finding all k occurrences of pattern P can be done in $O(n+k)$. Suffix trees can also be used to tackle many other problems (Gusfield; 1997), including the longest common substring by building a *generalized suffix tree* from concatenated strings (Vyverman et al.; 2012), as well as maximal exact matches using suffix links. Sparse suffix trees can be used to reduce memory, whereby only a subset of suffixes is stored (Kärkkäinen and Ukkonen; 1996).

Suffix array

A suffix array is an array A such that the ith element of A is the index of the ith lexicographically sorted (ith smallest) suffix of a string S. Li *et al.* (2018) provide an optimal algorithm for in-place suffix array construction (Li et al.; 2018), which can be performed in $O(n)$ time, provided the constraint that $|\Sigma| \in O(n)$. Suffix arrays are more space efficient than suffix trees (Manber and Myers; 1993), and can be used for computation of the BWT (Li et al.; 2018). For general alphabets, the suffix array can be found in $O(n \log n)$ (Franceschini and Muthukrishnan; 2007). In general, provided additional data structures, suffix tree algorithms can be replaced by suffix array algorithms (Abouelhoda et al.; 2004).

For general suffix tree algorithms performed with suffix arrays instead, several data structures may also be required (Abouelhoda et al.; 2004). The Burrows-Wheeler transform is related to the suffix array A by the fact that the ith element of the BWT array $B[i] = S[A[i] - 1]$. Similarly, the LCP table is an array whose ith element is the length of the longest common prefix of $S[A[i-1]:]$ and $S[A[i]:]$, that is, the length of the longest common prefix of strings indexed by consecutive elements of the suffix array. Both the LCP array and BWT array can be computed in linear time from the suffix array (Abouelhoda et al.; 2004). Examples of problems that can be computed by use of the suffix array and LCP array is finding maximal unique matches (MUM), such as in MUMmer (Delcher et al.; 1999). Querying the existence of a substring of length m can be performed in $O(m)$ time.

Strings referred to in the suffix array are prefixes of entries in the sorted table of cyclic permutations used in the BWT. As such, to compute the BWT from a suffix array, one must only find the character one position to the left of each suffix. E.g. if suffix array suffix i indexes into position j of S, that is $S[j:]$, then the ith element of the BWT is $S[j-1]$ (Li and Durbin; 2009d).

DA-FSA aka DAWG

A deterministic acyclic finite state automaton (DA-FSA), also known as a directed acyclic word graph (DAWG), is similar to a compressed prefix trie (radix tree), except that, informally, suffixes of words may rejoin other parts of the graph. In fact, a prefix trie is a type of DA-FSA. Formally, a DA-FSA is a deterministic finite state automaton accepting a language L with an acyclic transition function (Daciuk et al.; 2000). For a given language, a minimal acyclic finite state automaton (MA-FSA) is, amongst the DA-FSAs that accept that language, the one with the minimal number of states.

Assembly graphs

De Bruijn graphs (DBGs) are an essential data structure for several tasks in bioinformatics, including sequence assembly. Formally, we refer to the DBG of S as a tuple of vertices and edges (V, E), where each $v \in V$ is a k-mer, and $(i, j) \in E$ if $v_i = aW$ and $v_j = Wb$, where W is a substring of length $k-1$ (that is, the k-mers overlap by all but one base either side) (Rizzi et al.; 2019). For the more general overlap graph (OG), each v is sequence of any length, and $(i, j) \in E$ if there is any overlap between v_i and v_j. String graphs (SGs), introduced by Myers (Myers; 2005; Medvedev et al.; 2007) are OGs without edges that are 'transitively inferrable' (that is, by the transitive property, if $(x, y) \in E$ and $(x, z) \in E$, then $(y, z) \in E$; one of these edges can be inferred from the other two). OGs can be reduced to SGs by transitive reduction (Myers; 2005) in linear time. A usual assembly procedure is then to find paths within these graphs that are hoped to represent real biological sequences, termed *contigs*. The problem of finding a path in an OG that visits each read (node) exactly once, the Hamiltonian path problem, is intractable, although finding Eulerian paths in DBGs can be done in linear time (Pevzner et al.; 2001). However, in practice, the Eulerian superpath problem must be examined if one wants to incorporate read paths (and replace lost information), which is also NP-hard (Medvedev et al.; 2007). As such, assembly protocols often make use of heuristics. CELERA(Myers et al.; 2000) and CANU (Koren et al.; 2017) are examples of assemblers based on OLC. An example of a DBG assembler is EULER (Pevzner et al.; 2001).

Compact de Bruijn graphs

Compact representation of the De Bruijn Graph has previously been explored (Chikhi et al.; 2014). Methods include hash tables, as in AbySS (Ji et al.; 2011), sparse bit arrays (Conway and Bromage; 2011), lossy representation via subsampling k-mers or Bloom filters (where nodes are inserted into a BF) (Chikhi and Rizk; 2013), or minimizers (Chikhi et al.; 2014). Chikhi *et al.* (2014) developed a low-memory method for DBG construction with a trade-off of increased run-time. Explicitly storing nodes of a graph with k-mer labels, along with a hash map, DBGs for single large genomes can take hundreds of GB of space (Conway and Bromage; 2011). Since each node in a DBG has outdegree of at most $|\Sigma|$, DBGs can be stored efficiently by representation of nodes with 4 bits and appropriate compression (Conway and Bromage; 2011).

Colored de Bruijn graphs

Basic approaches to *de novo* assembly were not designed or well suited to accommodate biological variation not attributable to error (Iqbal et al.; 2012). Colored de Bruijn graphs (CBDGs), which are essentially DBGs with additional node colors (labels), were introduced for applications such as characterization of heterozygosity for a single individual or more complex variants such as deletions, which may also involve the use of a reference sequence as a path through the graph, and heuristics for identifying divergence from the path. CORTEX (Iqbal et al.; 2012) makes use of a hash table (with an integer k-mer representation as key), with 4 bits per node as previously described (Conway and Bromage; 2011) to encode edges, along with other information, packed into the value. Other methods include those based on Bloom filters, BWTs, or minimizers (Chikhi et al.; 2014). VARI (Muggli et al.; 2017) made use of a Bloom filter trie (Holley et al.; 2016) and BWT to implement CDBGs. In Rainbowfish (Almodaresi et al.; 2017), further compression was used by means of reducing redundant or repeated mappings. Almodaresi *et al.* (Almodaresi et al.; 2019) presented a method for compressing the CDBG, where color class labels may consume a large amount of memory.

Burrows-Wheeler transform

The Burrows-Wheeler Transform (BWT) is an invertible mapping that, in simple terms, permutes a string into a representation where similar characters are grouped together, and is used for string compression (Manzini; 1999). The BWT has found application in short read aligners such as Bowtie (Langmead et al.; 2009) and BWA (Li and Durbin; 2009d). Importantly, for mapping applications, a result of Ferragina and Manzini (2002) is key (Ferragina and Manzini; 2000). Firstly, as noted by (Li and Durbin; 2009d), any substring W of S must be present in an interval of the suffix array $A(S)$, because it is a prefix of a suffix, which are sorted in A. This leads to the following theorem from (the FM-index) (Ferragina and Manzini; 2000):

Theorem 1.3.2.1 *Let $i(W)$ be the first suffix array index for which W is a prefix, and $j(W)$ the last. Let $C(a)$ represent the count of bases of S that are lexicographically smaller than a. Let $h(a, k)$ be the count of a in $BWT[: k]$. Then:*

$$i(aW) = C(a) + h(a, i(W) - 1) + 1$$
$$j(aW) = C(a) + h(a, j(W))$$

And $i(aW) \leq j(aW) \Leftrightarrow W \in S$.

The functions C and h are referred to as the FM-index functions (Rizzi et al.; 2019). How does this work? In order to gain an intuitive understanding of this, consider each term: $C(a)$ locates, in the SA, the first position beginning with a, which is the start of a block that could contain the suffix aW; next, if there are any suffixes aW' smaller than aW, the number of occurrences of a in $B[: i(W)]$ gives the number of such prefixes, because for each such $W' < W$ starting at SA index k, $B[k] = a$. BWA-SW makes use of the algorithm of Hon *et al.* (Hon et al.; 2007) which allows for construction of the suffix array in $O(n)$ working space. The BWT, equipped with the FM index arrays, can be used to compute read overlaps (greater than some overlap distance ϵ (Simpson and Durbin; 2010)) in linear time (Rizzi et al.; 2019).

Filters

The Bloom filter (BF) (Bloom; 1970) is a commonly used data structures in bioinformatics (Holley et al.; 2016; Bradley et al.; 2019). Briefly, given an array M of m bits (initially zero), approximate membership query (and insertion) for an element x is performed by applying h hash functions, and testing whether all values $M[h(x)] = 1$. BFs allow for false positives but not false negatives, and can be used for querying large sequence datasets.

Counting quotient filters (CQFs), as Bloom Filters, are designed for approximate membership queries (AMQs). In both of these cases, although false negatives are impossible, there is some probability of a false positive, δ, specified by a parameter. As such, these data structures can be used to quickly pre-filter queries, before a slower exact query. CQFs are extensions of Quotient Filters (QF) (Bender et al.; 2012), which are compact hash tables. For the QF, false positives arise when two items have the same fingerprint (Bender et al.; 2012), that is, hash collisions (Geil et al.; 2018). A GPU-accelerated implementation of the QF has been designed (Geil et al.; 2018). Mantis (Pandey et al.; 2018a) makes use of a CQF, except instead of counts, k-mers are mapped to color classes instead, which is a method for storing a CDBG. CQFs are a superior alternative to Bloom filters (Pandey et al.; 2018a), such as in data locality (Geil et al.; 2018). CQFs may be used for k-mer counting applications. Similarly, advanced k-mer counting programs may make use of bloom filters and hash tables. SqueakR (Pandey et al.; 2018b) is an example of a k-mer counting tool that makes use of a CQF.

1.3.3 The seed-and-extend paradigm

The seed-and-extend method in bioinformatics is simple but powerful. In this method, small fragments termed seeds are matched first, and candidates are extended (Roberts et al.; 2004). This procedure is heuristic. For example, for local alignment, instead of the classic dynamic programming method that may be computationally infeasible, seed-and-extend can be rapid and often produce the same results. However, much thought goes into the design of the seeds. Two important methods are often used: minimizers and maximal repeats. Perhaps the most famous of a tool that makes use of seed-and-extend is BLAST (Altschul et al.; 1990).

Maximal repeats (MEMs and MUMs)

A repeated pair of substring indices is the tuple $((i, j), (k, l))$ such that $i \neq k, j \neq l$, and $S[i : j] = S[k : l]$, and is maximal (in left or right directions) if it cannot be extended (Abouelhoda et al.; 2006). A repeated pair is supermaximal if it is not a substring of any other maximal repeat. Maximal exact matches (MEMs) and maximal unique matches (MUMs) are both types of repeated pairs. a MUM is a supermaximal repeat that occurs in strings S_1 and S_2 exactly once each. For two sequences, a maximal exact matches (MEM) is a commmon substring that cannot be extended in either direction (Khan et al.; 2009). Finding MEMs is a crucial stage for many applications for comparing pairs of long sequences (Liu et al.; 2019). A few approaches can be used such as by building a generalized suffix tree (or array) out of both sequences, or by using minimizers (Almutairy and Torng; 2018). MEMs may perform best as anchor points for closely related sequences. A classic application of MUMs can be found in MUMmer (Kurtz et al.; 2004), which uses suffix trees, or in sparseMEM (Khan et al.; 2009), with sparse suffix arrays. Computation of MEMs is an active area of research (Liu et al.; 2019).

Calculation of MEMs is considered to be an important first step in many genome against genome comparison programs (Ohlebusch et al.; 2010). Several whole-genome alignment programs make use of MEMs or similar (Delcher et al.; 2002). MEMs can be computed from suffix arrays, with a few auxiliary data structures (Abouelhoda et al.; 2006), which allow top-down, bottom-up and suffix-link traversal. This fact was used to show every suffix tree algorithm can be replaced with a suffix array algorithm (Abouelhoda et al.; 2004). Key to this is the concept of the LCP array, and the LCP interval tree.

The typical approach for calculating MUMs involves the concatenation $S_1\#S_2$. However, it is also possible to 'stream' one sequence against another with a suffix tree for S_1 (Delcher et al.; 2002). Efficient implementations for finding MEMs and MUMs have been developed (Vyverman et al.; 2013; Khan et al.; 2009).

Minimizers

Minimizers are intelligently chosen k-mer seeds or 'fingerprints' (Marçais et al.; 2017). The computation of minimizers are also referred to as 'winnowing' due to development for document fingerprinting. Winnowing was introduced by Schleimer in 2003 for comparing the fingerprints of whole documents (Schleimer et al.; 2003). Naively, one can try to k-merize an entire sequence S, but in many cases, this may have large size. Alternatively, one can try to retain the every mth k-mer, but for many applications this is not permissible, since it means that the resultant fingerprint is sensitive to rearrangements, violating what Schleimer describes as 'position independence' (Schleimer et al.; 2003). The next approach was to take all hashes that are $0 \mod p$ for some p, an approach attributable to Manber (Manber et al.; 1994). Minimizers find use in many applications, from mapping (Li; 2016) to classification (Wood and Salzberg; 2014). Initially, Schleimer formulated two goals: i) guaranteeing that common substrings of length t are detected; ii) missing common substrings less than size k. The solution presented by Schleimer was to choose, in windows of size $w = t - k + 1$, the lexicographically smallest hash, which satisfies position independence. As with many algorithms in bioinformatics, different schemes exist, which may have variable performance depending on application (Marçais et al.; 2017; Almutairy and Torng; 2018).

1.3.4 Example applications: assembly and classification of RNA viruses

As is the case for many domains of bioscience, advances in sequencing technologies have enabled an unprecedented expansion of virus bioinformatics, although in some cases, collaboration between virologists and bioinformaticians has been slower than in other fields (Ibrahim et al.; 2018). Many of these advances have illuminated extreme diversity of viruses, with abundance up to 10 times higher than that of bacteria (Ibrahim et al.; 2018). This understanding has brought with it a renewed view of viruses not only as pathogens, but as critical components of many biological systems. However, many viruses can cause severe disease, which has never been more clear than with the rise of SARS-CoV-2. Many hundreds of thousands of RNA virus genomes are now available in online databases, such as GISAID (Shu and McCauley; 2017). However, virus data often presents unique challenges for algorithm and software design (Hölzer and Marz; 2017), when compared to other organisms.

Assembly

Since it is relatively well known that general purpose assemblers such as Velvet (Zerbino and Birney; 2008) or SPAdes (Bankevich et al.; 2012) can perform poorly on RNA virus sequencing outputs, often resulting in a large number of contigs (Baaijens et al.; 2019). Several *de novo* general purpose virus assemblers have been developed, such as VICUNA (Yang et al.; 2012), although in practice these may end up performing no better on a range of datasets. Some assembly pipelines have been developed that make use of conventional tools such as SPAdes with additional post-processing (Borges et al.; 2018). Next, we examine a few specific cases.

Assembly of viral quasispecies is well known to be problematic (Baaijens et al.; 2019; Rose et al.; 2016). For RNA viruses, an individual hosts a cloud of highly similar virus haplotypes, rather than a single genome, which makes assembly difficult. Even if reconstruction of haplotypes is performed accurately, estimation of their abundances is an equally difficult challenge; many approaches of this kind rely on a fixed reference. For many viruses, reliance on a reference genome may be acceptable; but for others, particularly where indels are more common, or diversity is extremely high (where many host species exist) it may not be in general. For a diverse virus like influenza, which infects many organisms and has a major reservoir in sea birds, reference-guideded assembly may be insufficient. In the clinical case, reference genomes may be acceptable, since most human infections are derived from well sampled lineages. However, for pandemic reassortants, this assumption may not hold. Three options exist for *de novo* assembly: general purpose assemblers; metagenomic assemblers; virus-specific assemblers (Hunt et al.; 2015; Baaijens et al.; 2019, 2017; Malhotra et al.; 2015). Of the general purpose assemblers, SPAdes was reported to perform the best for virus datasets (Baaijens et al.; 2019, 2017). Common metagenomic assemblers may not reconstruction variants at the strain level effectively for RNA virus datasets (Rose et al.; 2016). Some metagenomic assemblers have been designed specifically for bacteriophages (Antipov et al.; 2020). Whilst, on the surface these organisms may seem similar to RNA viruses, most bacteriophages have dsDNA genomes (Ackermann; 2009), and so present very different challenges. Similar, environmental sequencing for virus discovery (Alavandi and Poornima; 2012) presents different challenges.

Classification

Taxonomic classification is performed in several domains. Traditionally, taxonomic classification is performed on the basis of k-mers indexes (Menzel et al.; 2016). It has been argued that read mapping software is not appropriate for metagenomic classification due to the fact that, in metagenomic data, query sequences may have low identity to their reference (Menzel et al.; 2016). General purpose classifiers for microbiology include Kraken (Menzel et al.; 2016), Kraken2 (Wood et al.; 2019), and Kaiju (Menzel et al.; 2016). BLAST still represents one of the best methods when adapted to metagenomics, although it can be slow (Menzel et al.; 2016). Kraken is regarded as a fast but performant option, although classification is generally performed to the genus level (Menzel et al.; 2016). Kraken2 takes a similar approach, but, as in minimap2 (Li; 2018), uses only distinct minimizers in the query (Wood et al.; 2019). Often, classification of unassembled sequences can be advantageous, especially for metagenomic datasets. Kaiju (Menzel et al.; 2016) finds maximum exact matches with the BWT, working with protein translations due to lower variability, which the authors demonstrate to have superior performance to approaches based on k-mers. Finally, we note sequence classification methods have been developed for virus metagenome datasets (Simmonds and Aiewsakun; 2018).

1.4 Characterization of variants and intrahost virus populations

RNA virus infections are often characterized by some degree of intra-host variation which should be accounted for within bioinformatics pipelines. That is, instead of infection with a single strain, intrahost RNA virus populations are in fact genetically heterogeneous (Xue et al.; 2018). Influenza intrahost populations from a single source share similarities to those of coinfections between very similar strains. However, coinfection from two different sources, or subtypes, clades, and sub-clades, may often be distinguishable from normal populations. Furthermore, for influenza, intra-host variation is minimal, when compared to other RNA viruses such as HIV (Xue et al.; 2018; McCrone and Lauring; 2016), in principle due to the short duration of influenza infection. The number of variant sites across the entire genome with greater than one percent or so frequency is typically less than 15, with a proportion usually less than 10% (Debbink et al.; 2017; Xue et al.; 2018). In general, detection of coinfections is a special case of detection of mixtures. Mixture quantification, reconstruction of haplotypes, or variant calling from sequence data has been extensively studied. Here, I review many of these methods and some important preliminaries to aid in their understanding.

1.4.1 Mixture models: a short primer

Here, I briefly summarize some important aspects of mixture modelling, since it is an important foundation for several approaches used.

Finite mxture models

Mixture models are probabilistic models with a probability distribution function (density or mass) of the form $f(X|\theta, q) = \sum_{i=1}^{m} q_i f_i(X|\theta_i)$, where X is some datum, q_i is the ith mixture proportion of m clusters, $f_i(X|\theta_i)$ is some probability measure for the ith cluster with parameters θ_i (McLachlan et al.; 2019). If we sample from a mixture of sub-populations (termed components), each of which is individually homogeneous, then we may use a mixture model (Lindsay; 1995). A major challenge exists in correct choice of a number of clusters when it is not known *a priori* (Rufo et al.; 2010). In fact, finite mixture models are of deceptive simplicity. As will be described, several technical challenges arise which can complicate practical application of these models. A typical example is identifiability, which in the case of mixture models, is easily violated in a number of ways. Consider, for example, a mixture of two Gaussians $f(\mu_1, \sigma_1^2), f(\mu_2, \sigma_2^2)$. In this case, $(1/3)\mathcal{N}(0,1) + (2/3)\mathcal{N}(10,1)$ is not distinguishable from $(2/3)\mathcal{N}(10,1) + (1/3)\mathcal{N}(0,1)$, since they will have the same likelihood. In this sense, the likelihood function will not have a global maximum. This is worse for general finite mixtures of K components, for which there are $K!$ permutations that will evaluate to the same likelihood (McLachlan et al.; 2019). In the context of Bayesian MCMC, this is referred to as the 'label switching problem' (Jasra et al.; 2005). This kind of identifiability can be broken by the imposition of identifiability constraints (ICs), or in the case of Bayesian MCMC, other approaches (see Jasra *et al.* 2005). However, even with constraints, problems can arise, as will be described.

Dirichlet process mixtures

In non-parametric Bayesian mixture modeling, the number of clusters does not need to be assumed in advance (Görür and Rasmussen; 2010); non-parametric in here implies that there are, in a sense, an infinite number of parameters (Li et al.; 2019). Finite mixture models can be extended to countably infinite mixtures by the use of Dirichlet Process Mixture Models (DPMM). A Dirichlet process (DP) can be considered to be a distribution over distributions (Li et al.; 2019). Formally, a random probability distribution $X \sim DP(H, \alpha)$ if, for any finite partition of the sample space, $A_1, A_2, \ldots, A_n$,

$\Big(X(A_1), X(A_2), \ldots, X(A_n)\Big) \sim D(\alpha H_0(A_1), \alpha H_0(A_2), \ldots, \alpha H_0(A_n))$, where D is the finite Dirichlet distribution, H_0 is the base distribution, and $\alpha \in \mathbb{R}_+$ is a 'precision' parameter (Müller and Quintana; 2004). Alternatively, the DP can be formulated by $F(x) = \sum_{i=1}^{\infty} w_i \delta_{\mu_i}(x)$, where $\mu_i \sim H_0$, and the weights w_i are sampled recursively according to a stick-breaking process. For some intuition, consider for each sample X, probability mass randomly distributed across a countably infinite number of random points in the parameter space. For example, if $H_0 = \mathcal{N}(0, 1)$, then the positions are randomly distributed according to a standard normal, and the frequencies are randomly distributed according to a stick-breaking process. In the DPMM, the conditional probability of being assigned to existing cluster allows the construction of a Gibbs sampler (Görür and Rasmussen; 2010). A practical review of the DPMM can be found in (Li et al.; 2019).

As an example of how the DPMM can be practically applied, I will explain the approach and implementation performed by Zagordi *et al.* (2011) with their tool ShoRAH (Zagordi et al.; 2011), which was subsequently extended (Prabhakaran et al.; 2013). In their paper, these authors describe a haplotype as a $m \times 4$ matrix $\Theta = (\theta_1, \ldots, \theta_m)$, where each θ_i is a vector giving the probabilities of a categorical distribution describing the generation of bases at position i, which increases flexibility (as opposed to a fixed string). Aligned reads are then modelled as arising from a multinomial mixture with parameters Θ_k, where k is the mixture component. Using Dirichlet priors for Θ and π, the mixture proportions, the posterior distributions of $c_j|r, \theta, \pi$, $\theta_k|r, c, \alpha$, and $\pi|c, \gamma$, can be computed, where c is the vector of cluster labels, r are the reads, and γ, α are hyperparameters. This is made possible by the Multinomial-Dirichlet conjugacy: the posterior density of a parameter given a multinomial likelihood and Dirichlet prior is again Dirichlet (Holmes et al.; 2012). Sampling the cluster read membership variables has several advantages. Firstly, it allows fast Gibbs sampling of haplotypes since only reads that have been assigned to a given haplotype will contribute to it; because of this, counts at a given position can be considered instead of full reads.

Parameter estimation

Here, I assume a basic understanding of maximum likelihood estimation (MLE), as well as Bayesian Markov Chain Monte Carlo (MCMC) algorithms. It should be noted that high-dimensional parameter spaces can be difficult for both EM (Wang et al.; 2015; Yi and Caramanis; 2015) and MCMC (Norris and Da Silva; 2016). In general, this is a reflection of the 'curse of dimensionality' (Altman and Krzywinski; 2018). It may be desirable to reduce the feature space in order to improve performance and tractability (McLachlan et al.; 2019). However, Gibbs sampling may be particularly well equipped for handling high-dimensional parameter estimation (Chen, Tang and Li; 2019).

Expectation Maximization is an iterative method for finding maximum likelihood or MAP estimates of both parameters and hidden variables of a model (Dempster et al.; 1977). Although EM is considered to be cutting edge for fitting Gaussian mixtures (Dasgupta and Schulman; 2007), it can suffer from variable performance (Dasgupta and Schulman; 2007), notably depending on initialization (Melnykov and Melnykov; 2012), due to the presence of local minima in the likelihood function. Initialization can either be deterministic, such as with pre-clustering, or stochastic. In stochastic initialization, different starting positions can be chosen, and then the local maximum with the largest likelihood can be used (Melnykov and Melnykov; 2012). Although it has been claimed that EM is a sort of soft-clustering approach, in high dimensions, the cluster memberships are in practice hard due to concentration of the likelihood into a small volume (Dasgupta and Schulman; 2007). Additionally, size and shape regularization has been previously handled (Borgelt and Kruse; 2004; Yi and Caramanis; 2015).

One of the most difficult parts of finite mixture modelling is choice of the number of clusters. Several approaches have been described based on penalized likelihood critieria such as the AIC (see (Melnykov et al.; 2010) for a summary). Further, the standard likelihood ratio test result of Wilks (Wilks; 1938) is not valid in many cases because of technical problems with approximating the likelihood, including the null θ_0 being present on the boundary of the parameter space Θ, or even worse in the case of Gaussian mixtures, where the LR statistic is unbounded (Li et al.; 2009b). In fact, H_0 constitutes several lines in the parameter space Ω_0 (Lindsay; 1995): $\theta_1 = \theta_2$; $\pi = 0$; $\pi = 1$, which can be formulated as the the null $H_0 : \alpha(1 - \alpha)(\theta_1 - \theta_2) = 0$ (Li et al.; 2009b).

Since the usual $\mathcal{X}_d^2$ distribution of the statistic λ does not hold, alternatives have been developed, such as: bootstrapping (Feng and McCulloch; 1996; McLachlan and Khan; 2004); restriction of the parameter space (Chen and Cheng; 2000; Chen et al.; 2009); breaking the non-identifiability with the modified likelihood ratio (which can be interpreted as a MAP approach) (Chen et al.; 2001). It can be shown that, under some restrictive conditions, such as fixing all parameters except π, a type-II likelihood ratio test can be used, which results in the chi-bar-squared distribution (Lindsay; 1995), a mixture of the point mass at zero and a chi-squared with 1 degree of freedom. For a summary of commonly cited results regarding the LRT, see (McLachlan and Khan; 2004). For normal mixtures, simplifying the model or restricting the parameter space can make hypothesis testing more simple (Liu and Shao; 2004; Chen et al.; 2009). One approach is, essentially, to calculate the maximum likelihood parameters via EM, and bootstrap the likelihood ratio distribution (Feng and McCulloch; 1996). Each of these methods may find use in different contexts. Simple maximum likelihood may also find use. For example, after reconstruction of candidate haplotypes, Malhotra *et al.* (2015) (Malhotra et al.; 2015) use progressive removal until the likelihood decreases. The suitability of particular methods for different contexts may be questioned, however.

Many different approaches are used for detecting structure (which is fundamentally our task), including model selection, hypothesis testing, and classification. Although in many ways, these methods overlap, they are in fact suitable for different contexts. In the context of machine learning, classification can be defined as the process of determining class labels for test data given training data and labels (Aggarwal; 2014; Murphy; 2012). Clearly classification and hypothesis testing share similarities. In medical diagnostics, the relationship is clear (Pepe; 2003). In the context of decision theory, their similarities can be formalized (Parmigiani and Inoue; 2009; Wald; 1950).

1.4.2 Computational methods for RNA viruses

Taxonomic classification

Taxonomic classification in general involves the assignment of labels to sequences. For example, individual reads can be classified at the genus level. Several pipelines designed for generalized virus assembly perform taxonomic classification. For example, Genome Detective (Vilsker et al.; 2019) makes use of DIAMOND (Buchfink et al.; 2015) to identify viral reads, and place them into bins for subsequent assembly. In general, this problem can be decomposed into two parts. The first is accurate sequence alignment, which can be performed with several approaches, although BLAST is generally considered to be the gold standard (Buchfink et al.; 2015). However, BLAST is generally too slow for single reads (Buchfink et al.; 2015; Southgate et al.; 2020). VirAMP (Ajami et al.; 2018), designed for human virome classification also makes use of DIAMOND; however, in this case, the taxonomic level intended is coarse, with evaluation performed on a mix of several different virus families. The second task is quantifying how these reads distribute amongst taxonomic bins. This problem is non-trivial since many organisms may share large proportions of their genomes, and coverage may be low, which may result in false positives. An excellent case study of these problems can be found in the analysis performed by (Afshinnekoo et al.; 2015), where the authors claimed to have found *Yersinia pestis* and *Bacillus anthracis* on the New York subway. These claims were widely criticized (Ackelsberg et al.; 2015). General-purpose metagenomic taxonomic classification tools, such as Kaiju (Menzel et al.; 2016) or Kraken2 (Wood et al.; 2019), generally do not aim for strain-level classification. Lastly, classification often relies on *de novo* assembly, which can unsurprisingly lead to variable results (Sutton et al.; 2019).

Viral diversity estimation

Viral diversity can be studied on the scale of calling individual variants across an alignment, or globally, in terms of full haplotype reconstruction, or some combination of the two, which has been referred to as ‘local’ (Posada-Cespedes et al.; 2017).

Variant calling

Early variant calling approaches made use of either basic statistical models or *ad hoc* approaches (Howison et al.; 2019; Deatherage and Barrick; 2014; Wei et al.; 2011; Wilm et al.; 2012; Macalalad et al.; 2012; Koboldt et al.; 2009), often involving thresholding. Some methods previously applied to the study of intra-host populations do not make use of probabilistic models, but instead apply heuristic approaches to removing technical errors (Koboldt et al.; 2009)(Rozera et al.; 2009)(Archer et al.; 2010). Heuristic methods were often justified by the difficulties associated with distinguishing rare SNVs from errors. Although intrahost virus population may be more complex than sequencing data from other systems, this fundamental issue arises in several domains. As such, a number of approaches were motivated by, or designed for, human genome sequencing projects.

In general, variant calling is complicated by technical errors and sampling during the sequencing process, and has been studied extensively outside of virology (DePristo et al.; 2011). A common approach to resolve this issue is utilization of a statistical model, such as a binomial, in conjunction with a hypothesis test. In order to account for sequencing error distributions, sequencing of controls may be performed (Wang et al.; 2007; Gerstung et al.; 2012), or a constant specified error rate (Wei et al.; 2011). LoFreq (Wilm et al.; 2012) included the utilization of per-base Q-scores. Furthermore, alignment positions may be considered in pairs or higher-order combinations, in order to *phase* variants (Macalalad et al.; 2012; Yang et al.; 2013). Covama (Routh et al.; 2015) considers large matrices of all pairwsie sites and linkage disequilibrium. Furthermore, several computational tasks may be closely related to variant calling. For example, MAQ and SOAPsnp were designed with genotyping in mind (Li et al.; 2008, 2009c); Goya *et al.* (2010) presented SNVMix (Goya et al.; 2010) for tumour sequencing; as in other tools, data takes the form of sampled allelic counts. Ultimately, these tools compare a sequence data to a reference and aim to identify differences not due to error, although in general, genotyping may be tasked with inferring genotypes with higher ploidy, rather than a single variant as in a virus.

As an example, I briefly summarize the approach used by LoFreq (Wilm et al.; 2012). These authors use a Poisson-binomial to model the number of variant bases in a given alignment column. Unlike in the conventional binomial, which is composed of N i.i.d Bernoulli random variables, in the Poisson-binomial, each Bernoulli random variable can have a distinct success probability, which in this case is determined by the Phred score. P-values can be computed under the null hypothesis recursively.

McCrone *et al.* (2016) argued that it is very easy to overestimate viral diversity using common methods such as LoFreq (Wilm et al.; 2012) and DeepSNV (Gerstung et al.; 2012). This is easy to envision, since additional sources of error may be present on the basis of RT-PCR.

Haplotype reconstruction

The realization that considering single sites in isolation results in a loss of information allows insight into the entire haplotype sequences that may be present in a sample. Although tools such as V-Phaser (Yang et al.; 2013) may consider co-variation, this can be taken further. The goal of most haplotype reconstruction is to use this incomplete data to, in some way, estimate the haplotype sequences from which the reads are drawn. Two principle methods are used to this end: mixture modelling and graph algorithms. For pyrosequencing data, (Eriksson et al.; 2008) made use of error correction and subsequent haplotype reconstruction via read graphs and subsequent frequency estimation with expectation-maximization (EM). In order to do so, the authors find consistent paths through a read graph, and a set of haplotypes that explain the reads. By using their haplotype reconstruction algorithm to define a set of haplotypes for which frequency estimation is performed, the authors are essentially reducing the support of the haplotype distribution to a finite set. Since there are 4^L possible haplotypes for a given genome length L, reduction of the support to a subset of $m \ll 4^L$can be useful, computationally. If a haplotype is not consistent with any reads at all, then it will not have high likelihood. Similarly, ViSpA aims to assemble viral quasispecies and provide an estimate of their frequencies (Astrovskaya et al.; 2011), making use of a read graph in order to produce haplotype sequences, followed by EM.

ShoRAH (Zagordi et al.; 2011) employs a DPMM, error-correction by cluster centroids, and haplotype reconstruction (Zagordi et al.; 2010). Unlike the previous two methods, ShoRAH was designed for both 454 and Illumina Genome Analyzer sequencing reads. The authors make use of overlapping windows that are covered by some subset of the reads, for local reconstruction (error correction), followed by global reconstruction using EM, following the procedure described by (Eriksson et al.; 2008). Qure (Prosperi and Salemi; 2012) makes use of several herustic procedures, as well as probabilistic clustering similar to (Zagordi et al.; 2010), but do not actually make use of the Dirichlet process or Gibbs sampler. DPMM have found use in global reconstruction (Prabhakaran et al.; 2013); building on previous work, Prabhakaran *et al.* use a truncated DP mixture, but this time, employ an increasing sequence of nested windows, updating prior probabilities using the previous window, as the window grows.

A wide range of graph-based algorithms have been developed. HaploClique makes use of a read graph and maximum clique enumeration, where fully connected subgraphs, which represent groups of compatible reads, are extracted (Töpfer et al.; 2014). SdpR employs correlation clustering (Barik et al.; 2018). In general, these methods rely on construction of a graph, and some kind of partitioning or construction of paths based on a cost function. Some approaches assume an input set of ideal, error-corrected and aligned reads (Prosperi et al.; 2011). Other probabilistic methods make use of a first step where super-reads are constructed in order to make subsequent estimation tractable (Ahn and Vikalo; 2017).

Recent efforts have been dedicated to full *de novo* quasispecies assembly, such as Virus-VG (Baaijens et al.; 2018), SAVAGE (Baaijens et al.; 2017), PEHaplo (Chen et al.; 2018), MLEHaplo (Malhotra et al.; 2015). These methods tend to be graph-based; for example, Virus-VG constructs a *contig* variation graph (contig paths), and seeks to convert it to a genome-variation graph, where paths are full haplotypes. MLEHaplo relies on construction of candidate haplotypes from a de Bruijn graph, followed by maximum likelihood estimation via backward elimination (progressive removal of haplotypes until likelihood increases) (Malhotra et al.; 2015).

1.5 Fast nearest neighbor search and the edit distance

In virus pangenomics, especially in molecular epidemiology, after sequence assembly it may be desirable to query a sample against a database of hundreds of thousands of virus genome sequences in order to establish candidates for epidemiological linkage, or determine possible countries of origin for imported cases. This can be achieved by global alignment, or by faster methods for string comparison. Global alignment is a common computational challenge in bioinformatics that in general relies on dynamic programming for an exact solution. Furthermore, global alignment can form the core of subroutines within other bioinformatics software, such as multiple sequence alignment or read mapping (Li; 2013). Both global and local alignment can be framed in terms of *approximate string matching* (ASM), a common task from computer science (Navarro; 2001). As in alignment, ASM is usually divided into tasks; dictionary retrieval and substring matching. These are analogous to global and local alignment in bioinformatics. Due to its simplicity and historical usage in computer science, calculation of the edit distance tends to have faster algorithms than other alignment cost functions. Here, I briefly review these algorithms.

Approximate nearest neighbor and range search

One may desire to employ algorithms for neighbor search that make use of an alignment distance, analogously to spatial partitioning schemes. In general, two problems are faced when trying to use these algorithms: high dimensionality, and non-metric cost functions. For dimensionality greater than 10, exact k-NN algorithms often perform worse than linear scan (Ponomarenko et al.; 2014). For collections of hundreds of thousands of DNA sequences, each 30 kilobases in size, as with influenza or SARS-CoV-2, even filtering may be slow. For metric spaces with millions of sequences, navigable small world graphs (NSW) can be used (Malkov et al.; 2014; Malkov and Yashunin; 2018). In other domains, structures such as K-D trees (Bentley; 1975), or approaches based on locality sensitive hashing (LSH) can be employed. For near-metrics, which would be metrics except that they only satisfy the triangle inequality up to a constant multiplicative factor (including weighted global alignment distances), trees can also be used (Sahinalp et al.; 2003). However, for non-metric dissimilarity functions, the choices are few.

Nearest neighbor search for Levenshtein distance and Hamming distance have been researched extensively. Aside from bionformatics, searching for nearest neighbors in Hamming spaces arises in areas such as computer vision. One such example is multi-index hashing (Norouzi et al.; 2012). Conceptually, this relies on creating m hash tables for non-overlapping kmers, and calculating a bound on the number that must be matching in order for, say, a hamming distance of most r. For example, for a Hamming distance of 2, at most 2 kmers will mismatch. One problem is the presence of gaps or missing information, where a string with a gap could fit into 4 possible bins at that kmer position.

For approximate string matching, B-K trees (Burkhard and Keller; 1973) make use of the triangle inequality for searching discrete spaces, which were introduced due to the fact that traditional tree data structures such as in spatial partitioning break down in high dimensions, which may occur with long strings. However, in general, since there are no bounds on the degree of B-K tree nodes, speed-up can be variable depending on application.

Locality sensitive hashing (LSH) is an approximate method for clustering and nearest neighbor searches. LSH finds use in many areas of bioinformatics. Frequently, the technique is applied to sets of kmers in order to estimate the Jaccard index, a measure of set similarity (Marçais et al.; 2019), where kmer order is ignored. Marccais *et al.* (2020) introduced a form of LSH that also depends on order, in order to approximate the edit distance. An example of LSH is MinHash (Broder; 1997), which has found common use in MASH (Ondov et al.; 2016), although alternatives such as the HyperLogLog sketch also exist (Baker and Langmead; 2019). Again, missing data, such as poor coverage complicate these applications.

The edit distance

The edit distance, also known as the Levenshtein distance, is the number of operations required to transform one string into another (Navarro; 2001). Although many applications in bioinformatics employ more complex dissimilarity functions, for example which may define a matrix of substitution costs between amino acids, the edit distance is of key importance for big data applications because of special accelerated algorithms. Formally, as defined by (Levenshtein; 1966), let S_1, S_2 be two strings over an alphabet Σ. Then the edit distance $d(S_1, S_2)$ is equal to the minimum number of substitutions, insertions, or deletions required to convert S_1 into S_2, or vice versa. This distance is a metric (Levenshtein; 1966). Let the edit distance of substrings $S_1[: i]$ and $S_2[: j]$ be denoted as d_{ij}. An important property of the edit distance, that we will refer to as the recurrence relation (Gusfield; 1997), is as follows: for integers i, j, $d_{i,j} = \min(d_{i-1,j} + 1, d_{i,j-1} + 1, d_{i-1,j-1} + \delta_{ij}$, where δ_{ij} is 0 if $S[i] = S[j]$ and 0 otherwise. The *generalized edit distance* is an extension of the edit distance for which substitutions, insertions, and deletions can have variable costs. Other related dissimilarity functions are the Hamming distance (which is a special case of the edit distance), the q-gram or k-mer distance, and the block distance (Navarro; 2001). For the Hamming distance, $\alpha = k/m$ gives the error ratio (Navarro; 2001). If the cost function δ fulfills the triangle inquality and is strictly greater than zero then if an element is subjected to an operation it is not modified again (Wagner and Fischer; 1974).

Computing the edit distance with dynamic programming

The standard $O(n^2)$ dynamic programming algorithm for computing the edit distance is often called the Levenshtein (Levenshtein; 1966), Needleman-Wunsch (Needleman and Wunsch; 1970), or Wagner-Fischer (Wagner and Fischer; 1974) algorithm. A good exposition, with further developments, is given by Ukkonen (Ukkonen; 1985a). Let C_{ij} be the edit distance between $S_1[:i]$ and $S_2[:j]$, respectively. The core of these algorithms is the recurrence $C_{i,j} = \min(C_{i-1,j-1} + \delta(x_i, y_j), C_{i-1,j} + 1, C_{i,j-1}$. Note two important properties of this computation for the edit distance, $C_{ij} - C_{i-1,j-1} \in \{0, 1\}$ (*diagonal property*) and $C_{ij} - C_{i-1,j} \in \{-1, 0, 1\}$ and $C_{ij} - C_{i,j-1} \in \{-1, 0, 1\}$ (*adjacent property*). Another important property, the *cutoff* property, is that given $C_{ij} > k$, $\forall r > 0, C_{i+r,j+r} > k$, which allows us to abandon calculation should the threshold k be hit. Many variants for this algorithm have been developed with a range of cost functions. Ukkonen gives a straightforward method for bounding the number of computation by considering only diagonal strips of a given width in the middle of the alignment (Ukkonen; 1985a). In this case, the optimal alignment cannot be retrieved (Ukkonen; 1985a). It is important to know that C can also be computed row-wise, or even diagonally (Navarro; 2001). Ukkonen's algorithm was particularly useful for computation of whether the edit distance is within k. Landau and Vishkin (Landau and Vishkin; 1988) improved the worst case of this algorithm, and Landau *et al.* subsequently implemented incremental computation (Landau et al.; 1998). Wu *et al.* developed an $O(kn/\log n)$ algorithm, where k is the max distance, making use of a '4-Russians' approach. Importantly, Myers (Myers; 1999) introduced a $O(nm/w)$ bit-vector algorithm, where w is the machine word size. I next describe a few of these important algorithms. Many of these algorithms are described in terms of automata theory; in principle, an automaton for computing the edit distance may achieve an $O(n)$ search, but may be difficult to construct (Navarro; 2001). For approximate string matching, or edit distance computation with some practical bound k, $O(kn)$ runtime can be achieved (Ukkonen; 1985b).

Ukkonen's algorithms

In an important paper (Ukkonen; 1985a), Ukkonen improved on the basic algorithm to allow $O(s \min(m, n))$ time and space complexity for computation of s, the distance, derived from a method for verifying that $s \leq t$ in $O(t \min(m, n))$ time. These algorithms rely on the diagonals of the matrix d_{ij}. The latter emerges from a corollary 1 in (Ukkonen; 1985a): that if (i, j) (with score d_{ij}) lies on an optimal path between d_{00} and d_{mn}, then $-p \leq j - i \leq n - m + p$, where $p = \lfloor (1/2)(d_{mn}/\Delta - |n - m|) \rfloor$, and Δ is the minimum indel cost over characters (where we assume $m \leq n$, or we can reorder the strings such that this is true). In this case, only $1 + |n - m| + 2p \leq 1 + t/\Delta$ diagonals are evaluated, each with length $\min(m, n)$, so the algorithm is $O(t \min(m, n))$. The space complexity can be reduced as before. Furthermore, if one stores pointers to the leftmost and rightmost value $\leq t$ for each row, the band of diagonals can be shrunk as the distance increases, until they meet, with rejection; this does not change the worst case complexity. The edit distance can be computed from this procedure by iterate calls with increasing t; this calculation surprisingly has complexity $O(s \min(m, n))$ (Ukkonen; 1985a). Finally, the author presents another $O(s \min(m, n)), O(\min(s, m, n))$ algorithm, which we call the *diagonal algorithm*, that is in practice faster than the previously described iterative tests, if $\delta(a, b) = 1$ for all $a \neq b$. A good exposition of this algorithm is given in (Landau et al.; 1998). In short, it involves a variant of Dijkstra's algorithm (Dijkstra et al.; 1959) to greedily (hence it is referred to as the 'greedy algorithm') search for the shortest path between $(0, 0)$ and (m, n) (Landau et al.; 1998), which functions by incrementally extending diagonal stretches. Ukkonen's algorithm is $O(n + d^2)$ on average for random strings with $O(d^2)$ space for the alignment, and $O(d)$ space without. Furthermore, the requirements of the algorithm are that the match is zero and the mutations are positive (Powell et al.; 1999). This was also independently discovered by (Myers; 1986). Ukkonen's algorithm, whilst it is on average $O(n + d^2)$, can be improved to worst case $O(n + d^2)$ using preprocessed suffix trees admitting a LCA query, although this is reportedly not practical (Myers; 1986; Landau and Vishkin; 1988; Landau et al.; 1998).

Hirschberg's algorithm

Hirschberg's algorithm has $O(nm)$ time and $O(\min(m, n))$ space, and both calculates $d(S_1, S_2)$ and retrieves the operation sequence (Hirschberg; 1975; Powell et al.; 1999). The algorithm was originally devised for computing the longest common subsequence (LCS). The algorithm is fairly simple: it proceeds by splitting S_1 into two halves. Then, the standard DP algorithm of each half against S_2 is performed for both; the first is as the usual procedure, whereas the second is performed in reverse. Then, the correct partition of S_2 into two halves is found by considering the border where the two alignments meet. This can be stated in general by: if $d(S_1, S_2)$ is the optimal alignment of S_1, S_2, then for any partition $S_1 = A + B$, there is a corresponding one of $S_2 = C + D$ such that $d(S_1, S_2) = d(A, C) + d(B, D)$.

Bit Parallelism

Bit parallelism is exploited in several algorithms, the first of which was due to (Baeza-Yates and Gonnet; 1992), which is also known as the bitap algorithm, which has exact and fuzzy (computing neighbors within k) variants, for querying a small pattern P against text T. Let B be a table for which character c has bit mask $B[c] = b_m \ldots b_1$, with b_i set iff $P_i \neq c$. For example, $B(A; ATGAT) = 10110$. Let $D = d_m \ldots d_1$ (initially 1^m) store the state of a search, with $d_i = 0$ iff $P[1:i]$ currently matches the text (where a match is reported if $d_m = 0$). For each T_j, $D' = ((D \ll 1)|B[T_j])$. This algorithm was extended for regular expressions (Wu and Manber; 1992). One issue with basic bit parallel algorithms is that they are not well equipped to handle large patterns (Navarro; 2001). Wu and Manber (1992) extended the algorithm such that, for a basic automaton recognizing $L_k(P)$, each row is contained in a single word. Naturally, this is not applicable for large biological sequence patterns.

Wu *et al.* developed a sub-quadratic algorithm to build a NFA for recognizing edit distances based on the ‘Four Russians’ technique (Wu et al.; 1996). An important observation, as in Wu *et al.*, is that not only can we compute the difference matrix D (e.g. $C_{ij} - C_{ij-1}$ instead of C, but also that, for each cell at position (i, j), the value is determined by a finite number of values in adjacent boxes. Consider a ‘cell’, consisting of four coordinates $\begin{pmatrix} a & c \\ b & d \end{pmatrix}$ of C, where d is at (i, j). Assume that at the position of d, $P_i = T_j$ ($\delta_{ij} = 0$). We can ask the following question: given a, b, c, is $d < b$? This may seem obvious, and follow on from the basic DP rule. However, we can actually answer this question with only the differences $b - a$, $c - a$. For example, if $b > a$, then $d < b$. This yields a set of boolean equations. Although the required relations are described originally in (Myers; 1999), a clearer formulation is given by (Hyyrö; 2001; Hyyrö and Navarro; 2005). In order to handle $|P| > w$, blocks of size w are computed. In this case, the correct boundary conditions must be accounted for. Furthermore, this algorithm can be combined with Ukkonen’s banding cutoff algorithm (Hyyrö; 2003). This algorithm is also known as the bit-parallel matrix simulation (BPM) algorithm (Hyyrö; 2001; Hyyrö and Navarro; 2005; Šošić and Šikić; 2017).

Recent advances in the edit distance

Obtaining a subquadratic algorithm for computing the edit distance has been the focus of much research (Haeupler et al.; 2019). During the last 10 years there has been a significant amount of development in the field of approximating the edit distance in subquadratic time (Chakraborty et al.; 2018; Andoni and Nosatzki; 2020), although approximation factors better than 3 have not been achieved (Haeupler et al.; 2019). Several approaches have examined special cases for input strings, including: when input strings are compressible (Gawrychowski; 2012); when they are perturbed (Spielman and Teng; 2009); additional augmentation strings are supplied (Goldwasser and Holden; 2017). In particular, (Goldwasser and Holden; 2017) show how to compute EDIT (with other DP problems such as dynamic time warping) with correlated instances; in this case, the correlation between instances is exploited to solve the problem together. It should be noted that this is not the same as sequential learning. The approach of (Andoni and Krauthgamer; 2008) relies on a similar model for auxiliaries as in (Goldwasser and Holden; 2017). Other important applications are edit-similarity joins and edit distance metric embeddings (Indyk; 2001; Wang et al.; 2012; Gouda and Rashad; 2017; Kouckỳ and Saks; 2020; Zhang and Zhang; 2017, 2019).

LSH has also been applied for approximate similarity searches under the edit distance (McCauley; 2019). Algorithms that rely on preprocessing (Goldenberg et al.; 2020), or those based on automata theory have also been developed (Holub and Melichar; 2000; Mitankin et al.; 2011). Another interesting development is that of methods based on incremental learning (Breimer et al.; 2003). Other examples include: quantum algorithms (Boroujeni et al.; 2018); high-performance software libraries (Šošić and Šikić; 2017); as well as pre-alignment filters for short read alignment (Alser et al.; 2019); local alignment similarity joins (Wang et al.; 2017). Often, interdisciplinarity may be restricted for these problems (Wandelt et al.; 2014). Wendel *et al.* (2014) performed a large scale benchmarking of approximate search and join problems. Many of these approaches have pros and cons which depend on the precise datasets applied. As such, even if asymptotically, some approach may appear best, it may not be for all datasets.

Given many similar strings, computing the edit distance of each to a single query clearly may utilize many of the same computations. For many of these strings, in fact, in the standard DP algorithm, the optimal path through d may be nearly identical. Learning approaches have been applied to string edit distance (Ristad and Yianilos; 1998).

Within the field of bioinformatics, the edit distance is regularly used in recent research, particularly in mapping applications (Alser et al.; 2017). Recent developments include: fast libraries for edit distance calculation, where sequences of length a million have run-time on the order of seconds to minutes (Šošić and Šikić; 2017); LSH for the edit distance (Marçais et al.; 2019).

Indexing

Tries or their compressed analogues (Hanov; 2013; Lu et al.; 2014; Gouda and Rashad; 2017; Qin et al.; 2019), or a MA-FSA (Hanov; 2013; Daciuk et al.; 2000) can be used to first index sequences. In the tree-based approach, we do not need to perform the same computation for every common prefix in the references; instead, we an save components of the desired matrix and reuse them. This approach is hard to find in academic literature, but is often credited to a blog by Steve Hanov (Hanov; 2013; Waddington; 2016; Saluja et al.; 2017; Fahda and Purwarianti; 2017). However, in Hanov's original implementation, the naive $O(nm)$ DP is used. As it turns out, Ukkonen developed a very similar version of this algorithm designed for substring search, which made use of a suffix tree (Ukkonen; 1993). Furthermore, similar approaches have been developed previously for autocompletion and neighbor joins (Chaudhuri and Kaushik; 2009; Ji et al.; 2009). A large component of optimizing tree-based methods is selection of which nodes to compute (Qin et al.; 2019), and strategies for filtering nodes.

Filtering for the edit distance

Since computation of the edit distance is so expensive, many efforts are directed first towards filtering (Lu et al.; 2014). With a smaller pool of candidate neighbors, the search is quicker. Several approaches exist to this end, including those based on n-grams (Lu et al.; 2014), tries, or B+ trees (Lu et al.; 2014). q-gram approaches simply require a sufficient number of q-grams to be held in common with a query, using inverted lists. Furthermore, q-gram based approaches may require a much larger amount of memory, depending on the value of q. Q-grams were originally introduced by Ukkonen (Ukkonen; 1992), can be computed in $O(m + n)$, and provide a lower bound for the edit distance.

1.6 Trees and molecular clocks for RNA viruses

Once sequences have been assembled and grouped, phylogenetic analysis usually follows. Molecular clock inference and divergence dating has become a key process in molecular epidemiology and the study of pathogens. Many examples exist in the literature of their application. For example, using molecular sequence data, (Streicker et al.; 2010) saught to quantify per-capital cross-species transmission rates for rabies virus between North American bats and humans, and (Robbins et al.; 2003) estimated the date the HIV-1 epidemic began in the US, and also inferred population history. For an overview of molecular clock methods see, including many practical considerations, see (Ho and Duchêne; 2014; Kumar and Hedges; 2016). For a review of computational optimization, see (Stamatakis; 2019; Guindon and Gascuel; 2019).

1.6.1 The tree likelihood

DNA substitution models

Not only are DNA substitution models at the computational core of phylogenetic estimation, and modern molecular clock methods as well, substitution parameter estimates can affect dating (Shapiro et al.; 2006; Schenk and Hufford; 2010)(Posada; 2001). Substitution models have been developed for over 40 years (Arenas; 2015), and include the Jukes-Cantor (JC) (Jukes and Cantor; 1969), Hasegawa-Kishino-Yano (HKY) (Hasegawa et al.; 1985), and General Time-Reversible (GTR) (Tavarè; 1986) models. Additionally, specification of Gamma-distributed rate heterogeneity (Yang; 1994) and proportion of invariant cites (Shoemaker and Fitch; 1989) can be incorporated into these models (Arenas; 2015). Correct choice of substitution models is an important step in phylogenetics pipelines (Arenas; 2015). For example, The HKY model (Hasegawa et al.; 1985) of DNA evolution assumes finite, independent sites, with i.i.d substitution events. Let a DNA string be $(x_1, x_2, ..., x_s)$, with 4^S possible states. The model assumes that the evolutionary process is Markovian with:

$$\frac{d}{dt}P(t) = P(t)Q$$

Where Q is the infinitesimal generator. Depending on the model used, Q has a varying number of parameters. For example, in the HKY model:

$$Q_{ij} = \begin{cases} \alpha\pi_j & \text{transition} \\ \beta\pi_j & \text{transversion} \\ -\sum_{i\neq a}\alpha\pi_a - \sum_{i\neq b}\beta\pi_b & i = j \end{cases}$$

where a are bases that can transition to i, and b are bases that can transvert to i. The General Time Reversible (GTR), was developed by (Tavarè; 1986), which consists of 6 transition rate parameters, and a stationary distribution.

Felsentein's algorithm

The fundamental core of many phylogenetics analyses requires evaluating the tree likelihood (Felsenstein; 1981), also known as the phylogenetic likelihood function (PLF), which is known to be a computationally demanding task. As in Felsenstein's original 1981 paper (Felsenstein; 1981), sites are typically treated as independent, so likelihood of trees is computed at individual sites and their product taken. Assume a tree topology $\mathcal{T}$, branch lengths $\vec{b}$, and a substitution model with parameters θ.

Let $P_{ij}(t)$ be the Markov transition probability at time t. If the ancestral states are known, then we can calculate both the conditional probability $P(D|\mathcal{T}, S)$ or the joint probability $P(D, S|\mathcal{T})$. It may now become clear that the states S are, in a sense, hidden or latent variables. Briefly, recall:

$$P(A, B, C) = P(A)P(B|A)P(C|B, A)$$

But if, for argument's sake, $C \perp\!\!\!\perp B$ conditional on A, such as in a phylogenetic tree then:

$$P(A, B, C) = P(A)P(B|A)P(C|A) = P(A) \prod_{X \in \{B,A\}} P(X|A)$$

In general, for random variables with tree-like structures, we can therefore exploit the structure of the variables (as we will see, importantly for improving run-times) to define useful recurrence relations. Since most of the time the ancestral states are not known, they must be integrated out. For n interior nodes, we will have 4^n possible internal states at each locus. However, due to the structure of the tree, the likelihood for each site can be computed in $O(n)$. Felsenstein's tree likelihood can be stated for general node states. Firstly, note, if the root state is not known:

$$L(\mathcal{T}|D) = P(D|\mathcal{T}) = \int P(D|\mathcal{T}, s_0)P(s_0)ds_0$$

Furthermore, we define the quantities, for each node n with state s:

$$L_s^{(n)} = P(D|\mathcal{T}, s) = \prod_{i \in \mathcal{C}_n} \int P(D_i|\mathcal{T}_i, s_c)P(s_i|w)ds_i$$

Where D_c is the data under child node c, and $\mathcal{T}_c$ the subtree rooted at node c. Here the integral is general, and for discrete states corresponds to a sum. The choice of root state is arbitrary for the purposes of calculation. For integrating out ancestral states, in Felsenstein's original formulation, for discrete nucleotide character states, this was:

$$L_s^{(n)} = P(S_n = s, D_n|\mathcal{T}_n, \vec{b}_n) = \prod_{i \in \mathcal{C}_n} \Big(\sum_{s_i} P_{s,s_i}(b_i) L_{s_i}^{(i)} \Big)$$

Note that this formula assumes both the topology and the branch lengths are known. Computing the maximum likelihood branch lengths can be performed numerically as usual. However, this is slow, especially since the full likelihood requires $O(nL)$, where L is the alignment length (although some heuristics can accelerate this). Derivatives for the tree likelihood can be computed to make use of conventional numerical optimization algorithms where required (Schadt et al.; 1998; Ji et al.; 2020).

Given the above, it is possible to directly calculate the maximum marginal likelihood of a state for the root. However, it is also possible to compute the ancestral states that maximize the posterior $P(s|D,\mathcal{T})$ in linear time with dynamic programming (Pupko et al.; 2000). For example, for reconstruction of ancestral sequence states, let be the $C_n(s)$, the maximal reconstruction of the subtree at n given a parental state s, and $M_n(s)$ the corresponding likelihood. Then, given suitable initial conditions at the leaves (see (Pupko et al.; 2000) for details):

$$M_n(s) = \max_j P_{ij}(t_n - t_p) \prod_{c \in \mathcal{C}} L_c(j)$$

$$C_n(s) = \arg\max_j P_{ij}(t_n - t_p) \prod_{c \in \mathcal{C}} L_c(j)$$

Why does this work? Consider a tree $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$. Let w be a maximal reconstruction of the states of $\mathcal{T}$. Let n be the node in $\mathcal{T}_1$ that joins $\mathcal{T}_1$ and $\mathcal{T}_2$. Let s be the state of n in the maximal reconstruction. Then the states of $\mathcal{T}_2$ must be the same as the maximal reconstruction of $\mathcal{T}_2$ conditional on s at n. In theory, for other node states, the relationship holds.

Optimization of the PLF

High performance libraries have been developed in C++ for likelihood calculations (Flouri et al.; 2015; Ayres et al.; 2012), amongst other phylogenetic tasks. Calculation of independent sites can be parallelized trivially, although in computation, faster methods may be utilized (Stamatakis; 2019). Furthermore, optimization of model parameters is challenging. Optimization of branch lengths, in particular, can be complicated by local maxima (Chor et al.; 2000). Typically, in optimization, many of the likelihood vectors associated with nodes do not need to be recomputed (Stamatakis; 2019), and the principle means by which this can be achieved is by collapsing duplicate patterns in the tree across sites down to a single representative; this is known as the site repeats (SR) method. The simplest type of site repeat is when entire columns of a MSA are identical. However, subtrees may also be identical between sites. These can be efficiently identified (Kobert et al.; 2017). Efficient parallelization with SR has been explored (Morel et al.; 2017). Furthermore, for nodes with two child tips, or one child tip and one internal tip, likelihood vectors can be precomputed. x86 vector instructions are also commonly explored. Several open theoretical problems exist. External memory algorithms have also been developed (Izquierdo-Carrasco and Stamatakis; 2011) for when memory requirements are problematic.

1.6.2 Methods for molecular clock estimation

The molecular clock is often first credited to Zuckerkandl and Pauling (Zuckerkandl and Pauling; 1962). Although the use of this phrase may refer in particular to a strict molecular clock, where the substitution rate is constant, due to the development of relaxed models, we refer explicitly to a strict clock when the rate is assumed to be constant. Furthermore, the rate defined here is distinct from the biochemical mutation rate; mutation rate is the error rate during replication, whereas substitution rate is the rate of spread and fixation of new mutations, although they may be equal under some assumptions including neutrality (Drummond et al.; 2003). The relationship between mutation rate and evolutionary rate is complex. Although under neutrality, evolutionary rate should be a linear function of mutation rate, this is not the case when neutrality is violated (Sanjuán; 2012). Clock rate variations may be caused by horizontal gene transfer, such as recombination (Schierup and Hein; 2000), or positive selection (Wróbel et al.; 2006). In terms of positive selection, evolutionary rates can vary even over the course of a single infection. Neutral sites can be examined in order to attempt to control for positive selection (Wróbel et al.; 2006). As a consequence of this, we might ask: is it possible to accurately date sequences on short epidemiological time scales? In analysis of 50 RNA viruses statistically significant rate variation was found by (Jenkins et al.; 2002). Rates of evolution for rabies virus varied between species in different geographical ranges (Streicker et al.; 2012), possibly due to variable seasonality and climate-associated transmission. For closely related viruses, host factors may result in variable evolutionary rates, particularly those factors that influence transmission or replication (Streicker et al.; 2012). Furthermore, the mutation rate itself can vary between hosts (Combe and Sanjuan; 2014). So, rates can vary, and as such, molecular clocks should be informed for each dataset, rather than assumed from previous studies. It should be no surprise then, that for influenza, a virus with animal reservoirs, reassortment, complex patterns of selection, seasonal variation, and so on, that molecular clock rates would vary across branches. Strict clock rates may be more accurate for shallow phylogenies (Brown and Yang; 2011). This makes sense because the time-scale during which selection and horizontal gene transfer can act is reduced. The observation that clock rates are also estimated to be higher for small-scale outbreaks has been suggested as possibly a result of purifying selection (Möller et al.; 2018; Woodhams; 2006; Ho et al.; 2005).

In general, molecular clock estimation methods can be categorized as regression or distance-based, maximum likelihood, or Bayesian (Drummond et al.; 2003). Early strict molecular clock estimation methods relied on simple linear regression (Gorman et al.; 1990) or comparing pairs of sequences (Li et al.; 1988). For example, where a pairwise method was used to estimate rates in different regions of HIV proteins using a fixed tree (Li et al.; 1988). With this method, if t is the time between isolation of two sequences, a the (constant) clock rate, t_i and l_i the time and branch length between common ancestor of each i in a pair, then $l_2 - l_1 = at_2 - at_1 = at$. Also, if outgroup is $i = 3$, $l_2 - l_1 = d_{23} - d_{13}$. So $a = (d_{23} - d_{13})/t$. Although t_1 and t_2 are not known, their difference is. Also, $l_2 - l_1$ could be calculated from a fixed tree, or from divergence estimates to an outgroup. So, with $l2 - l1 = a(t2 - t1) = at$. Intuitively, this method works because there is an extra time period between sampling of the taxa; a clock rate estimate is the difference in length over the difference in time. Methods based on mean path lengths (Britton et al.; 2002) can also be used, where the mean distance between a calibration node and its terminal (Rutschmann; 2006). Least squares can also be used to provide an approximate solution (To et al.; 2016).

For probabilstic methods that assume a Markov model of substitution, Felsenstein's pruning algorithm (Felsenstein; 1985) can be used directly for clock estimation (Rutschmann; 2006)(Rambaut; 2000) for taxa with non-contemporaneous tips. Methods based on UPGMA were also developed as an extension of the pairwise method (Drummond and Rodrigo; 2000). Root-to-tip regression methods are flawed precisely because taxa are not statistically independent (Drummond et al.; 2003). Pair-wise linear regression methods, such as those used by Drummond and Rodrigo (2000) (Drummond and Rodrigo; 2000), make use of the fact $\Theta = 2N_e\mu_g$, where N_e is the effective population size, and μ the mutation rate (per site per generation), and $E[d_{ij}] = \mu|t_i - t_j| + \Theta$. Generalized least squares methods account for non-independence of samples, and assume time itself is a random variable (Drummond et al.; 2003). In a Bayesian setting, as a probabilistic method, similar substitution models can be employed to maximum likelihood methods. In some contexts, such as with the software BEAST, complex models that incorporate molecular clocks, substitution models, and demographic tree priors can be simultaneously estimated (with computational cost). In this case, and others where tree priors are used, they can affect molecular clock estimates (Möller et al.; 2018). Treedater (Volz and Frost; 2017) makes use of the Langley-Fitch model (Langley and Fitch; 1974), which assumes a low mutation rate and large genome such that reversions are rare, with Poisson-distributed substitutions across branches. In order to account for rate variation (overdispersion), Volz and Frost make use a gamma-Poisson (Negative Binomial) distribution.

Treetime (Sagulenko et al.; 2018) employs a method similar to expectation-maximization by iteratively optimizing ancestral states followed by branch lengths, since computationally, for short branch lengths, each can be optimized easily conditionally on the other one. Branch lengths and ancestral states are iteratively, alternatively optimized, similar to as in EM. The authors make use of the fact that, if ancestral state reconstruction is performed first, branch lengths can be maximized in linear time.

If evolution proceeds under a strict molecular clock, then the resultant tree should be ultrametric; that is, the genetic distance between any two isochronous sampled taxa and their MRCA should be equal, and as such all tree leaves should be at the same position. Tests were developed early for the hypothesis of strict a molecular clock (Tajima; 1993). Likelihood ratio tests were also applied to heterochronous samples (Rambaut; 2000); in this case, the likelihood of the strict clock can be compared to the likelihood of an unconstrained tree by the usual statistical procedure. Bootstrapping and jackknifing can also be performed, which may not require independence of samples. With parametric bootstrapping, data is simulated under the inferred model, and with non-parametric, data is subsampled, though this should be performed on nucleotide sites, not root-to-tip distances (Drummond et al.; 2003). Likelihood-based methods that use Felsenstein's likelihood are more accurate and sensitive, and admit likelihood ratio tests (Drummond et al.; 2003). Similarly, multiple rates dated tips (MRDT) models allow stepwise changes in substitution rates, and can also be subjected to likelihood ratio tests (Drummond et al.; 2003). Since the strict clock, although not the most realistic, is simple and computationally easy to estimate, the development of tests for the strict clock hypothesis is an active area of research (Antoneli et al.; 2018).

For autocorrelated relaxed clocks, the evolutionary rate itself has an evolutionary rate (Gillespie; 1994). These models generally could be partitioned into those where rates are autocorrelated amongst branches, that is, rates are inherited ancestrally, or per-branch rates are independent, and constrained by some distribution, (Rutschmann; 2006). Autocorrelated relaxed molecular clocks were introduced, in a Bayesian setting with (Thorne et al.; 1998). In this case, the number of substitutions (or generally events) is Poisson with rate $B(T) = \int_0^T R(t)dt$. With Felsenstein's likelihood, a separate rate is estimated for each branch, which corresponds to an 'unrooted' tree (in the absence of an outgroup) (Drummond et al.; 2006). Examples of models of rate evolution include lognormal, gamma, exponential, and the Orstein-Uhlenbeck (OU) process (Aris-Brosou and Yang; 2002). In the lognormal model (Thorne et al.; 1998), discrete rates are drawn from a lognormal centred at the rate of the ancestor of a given branch. Others include the Cox process (Cutler; 2000). Alternatively, for 'local' clock models, branches are partitioned, and given specific rates (Aris-Brosou and Yang; 2002); for example, if one knows that two parts of a tree would have different rates, then two rates can be modelled, rather than one. For example of a rigorous comparison of clock rate evolution, see (Aris-Brosou and Yang; 2002). The relaxed molecular clock can be estimated in conjunction with integration over trees (Drummond et al.; 2006). Furthermore, random local clocks can also be estimated in this manner (Drummond and Suchard; 2010): 2^{2n-2} random local clock change points can be estimated in a Bayesian setting, and, simultaneously, the posterior probability of a single rate, i.e. a strict clock, can be evaluated.

Clearly, the estimation of molecular clock rates is complex, and often computationally difficult. As such, naive application of methods to epidemiological research could easily result in poor estimates. Furthermore, specific application domains may require specific methods or parameters. For example, BactDating was developed and benchmarked with a two step approach, with conventional phylogenetic tree estimation and subsequent dating, allowing for signals of recombination to be assessed in between (Didelot et al.; 2018). Interestingly, one of the main benefits of BactDating is cited as the use of a fixed tree, though it should be noted that fixed trees are possible with BEAST. It is not uncommon for studies to make use of a fixed tree for clock-rate estimation. In some cases, the number of mutations across a branch is assumed to be Poisson (Volz and Frost; 2017; Huelsenbeck and Ronquist; 2001); in other cases, the non-integer nature of estimated tree branches motivates the use of a Gamma distribution with equal variance (Didelot et al.; 2018).

1.7 Experimental objectives

This thesis is divided into two parts; one focused on aspects of sequence reconstruction, and one focused on aspects of analysis. The first part is comprised of the following objectives:

1. Typically, in sub-routines that involve mapping, fixed references are utilized; often, these are old or arbitrarily chosen sequences. **I aimed to demonstrate complications involved in hard-coding RNA virus reference sequences, and develop an algorithm for intelligent selection of influenza references to minimize bias**.

2. In large sequencing initiatives, cross-contamination of samples can occur. Furthermore, biological coinfections can result occur in influenza infection. A large body of work exists for the characterization of population structure in virus WGS reads. **I aimed to show that this work can be leveraged for the automated detection of mixed samples for sequencing pipelines**.

3. In routine epidemiology, retrieving the nearest neighbors for a sequenced sample is a routine task used as a sub-routine in various activities, including typing, phylogenetics, phylodynamic models, and the characterization of importation events. Often, this relies on a pseudo-alignment or a slow multiple sequence alignment (MSA). **For the final chapter in this section, I adapted fast algorithms for calculation of the edit distannce to calculation of the SNP distance, for exhaustive search of huge reference datasets**.

The second part of this thesis is focused on RNA virus phylogenetics and phylodynamics. In particular:

1. Molecular dating of RNA virus sequences has become well established over the last decade; however, the success and accuracy of various methods can depend on the information present in the sequences, which is a function of mutation rates and time-scales, amongst other variables. **I benchmarked several tools for molecular dating of samples from simulated influenza epidemics, and examined their applicability in routine epidemiology**.

2. Phylodynamic modelling allows inference of population parameters from observable patterns in trees, which has found increased use in public health surveillance. **I used phylodynamic methods to characterize the exponential growth of SARS-CoV-19 during the first wave of the COVID-19 pandemic in wales, and evaluate the difficulties in this approach for routine epidemiology**.

3. Lastly, **I applied simple ancestral state reconstruction methods for the isolation of imported lineages into Wales during the COVID-19 pandemic and examined signatures of importation, as well as geographical mixing. Unlike more sophisticated methods, I argue that simple methods could be automated in future**.

Part I

Optimized software for influenza virus whole-genome sequencing pipelines

Chapter 2

Influenza virus reference selection from short read data

A derivative of this work has been submitted to bioRxiv.org as a pre-print, and later published in *Bioinformatics*. As presented, the writing and work of this chapter was carried out by Joel Southgate, including algorithm development and analysis. Laboratory work was carried out by collaborators as indicated in authors' contributions. Intellectual inputs are also indicated in authors' contributions.

2.1 Abstract

2.1.1 Background

Influenza viruses represent a major public health burden worldwide, resulting in an estimated 500,000 deaths per year, with potential for devastating pandemics. Considerable effort is expended in the surveillance of influenza, including major World Health Organization (WHO) initiatives such as the Global Influenza Surveillance and Response System (GISRS). To this end, whole-genome sequencing (WGS), and corresponding bioinformatics pipelines, have emerged as powerful tools. However, due to the inherent diversity of influenza genomes, circulation in several different host species, and noise in short-read data, several pitfalls can appear during bioinformatics processing and analysis.

2.1.2 Results

Conventional mapping approaches can be insufficient when a sub-optimal reference strain is chosen. For short-read datasets simulated from human-origin influenza H1N1 HA sequences, read recovery after single-reference mapping was routinely as low as 90% for human-origin influenza sequences, and often lower than 10% for those from avian hosts. To this end, I developed software using *de* Bruijn

Graphs (DBGs) for classification of influenza WGS datasets: VAPOR. In real data benchmarking using 257 WGS read sets with corresponding *de novo* assemblies, VAPOR provided classifications for all samples with a mean of $>$99.8% identity to assembled contigs. This resulted in an increase of the number of mapped reads by 6.8% on average, up to a maximum of 13.3%. Additionally, using simulations, I demonstrate that classification from reads may be applied to detection of reassorted strains.

2.1.3 Conclusions

The approach used in this study has the potential to simplify bioinformatics pipelines for surveillance, providing a novel method for detection of influenza strains of human and non-human origin directly from reads, minimization of potential data loss and bias associated with conventional mapping, and facilitating alignments that would otherwise require slow *de novo* assembly. Whilst with expertise and time these pitfalls can largely be avoided, with pre-classification they are remedied in a single step. Furthermore, this algorithm could be adapted in future to surveillance of other RNA viruses. VAPOR is available at `https://github.com/connor-lab/vapor`. Lastly, VAPOR could be improved by future implementation in C++, and should employ more efficient methods for DBG representation.

2.2 Background

Influenza virus WGS in routine surveillance poses several challenges. Firstly, the RNA genome of the influenza virus, as with other RNA viruses, is diverse and mutable, which can result in genome divergence on a yearly basis; this process also leads to *antigenic drift*, which is the primary mechanism whereby strains avoid natural or vaccine-induced immunity (Taubenberger and Kash; 2010) (Petrova and Russell; 2018). Secondly, due to the presence of several intermixing host species, principally swine and birds in addition to humans, novel pandemic strains can emerge by the process of *reassortment*, which occurs when one or more of the 8 RNA segments of the influenza genome are exchanged (Bouvier and Palese; 2008), resulting in a new virus that has a genome of mixed segments. These processes can both present a challenge to epidemiological surveillance and cause major public health crises; as such, it is crucial that bioinformatics approaches utilized with influenza virus datasets are robust.

Despite the increasing application of Next-Generation Sequencing (NGS) to influenza, the pitfalls associated with current bioinformatics approaches have not been explored in depth. Influenza virus assembly poses additional challenges due to biological population complexity and additional error resulting from RT-PCR(Orton et al.; 2015). Firstly, I aim to provide evidence that current mapping approaches can, due to diversity of influenza genome sequences, result in unmapped reads, which can potentially result in data loss and bias in sequences that are subsequently recovered, analyzed, and submitted to public databases. This has been previously noted in other RNA viruses (Wymant et al.; 2018). Whilst alternatives, such as read classification by mapping to a large database of influenza sequences (Yu et al.; 2014) and subsequent *de novo* assembly can help to resolve this issue, such pipelines are often complex, slow, and require expertise that is not necessarily available in routine surveillance or public health laboratories. Secondly, even if bioinformatics pipelines are chosen judiciously, sequences of zoonotic origin may fail to be identified, resulting in a dataset that appears to be low coverage, missing segments, or missing potential future pandemic reassortments. Furthermore, even with recent assembly programs, misassembly can occur (Wymant et al.; 2018).

I aim to show that this problem can be resolved by classification of isolates directly from reads prior to analysis by directly querying a De Bruijn graph (DBG) built from the reads. Mapping reads directly to a DBG has been previously argued to be less biased than that of mapping to assembled contigs (Limasset et al.; 2016). Directly querying DBGs instead of assembled sequences has been previously addressed (Limasset et al.; 2016)(Holley and Peterlongo; 2012)(Liu et al.; 2016)(Salmela and Rivals; 2014), although examples focus on mapping reads to a DBG. To my knowledge these approaches have not been applied to pathogen classification from reads. Instead of mapping reads to a DBG, I sought to further develop a simple method for querying short influenza genome sequences against a short read DBG in order to retrieve the most similar reference for mapping applications. In doing so, I leverage the large number of publicly available influenza segment sequences. I compare a tool that implements this algorithm, VAPOR, with both slow BLAST-based (Altschul et al.; 1990) and fast k-mer-based MASH (Ondov et al.; 2016), and show superior or equivalent results in several use cases with reasonable run-times. I show through simulation, that given a set of influenza reads, possibly contaminated with human or bacterial sequences, a highly similar strain in the NCBI influenza virus resource (NIVR) database (>20,000 strains) can be selected, achieving reasonably fast near-strain-level classification.

2.3 Methodology

2.3.1 WGS datasets

Total RNA was extracted from patient samples using the NucliSens easyMAG instrument according to the manufacturer's instructions. Following RNA extraction, a one-step RT-PCR (Quanta biosciences qScript XLT kit, following manufacturer's instructions) was then undertaken to generate DNA for sequencing using the primers previously described for influenza A (Zhou et al.; 2009) and influenza B (Zhou et al.; 2014). Sequencing was performed using Illumina sequencing instruments. Libraries were prepared using NexteraXT, and samples were then multiplexed for sequencing. Samples were run on a MiSeq (2x250bp V2 kit 44 samples) and NextSeq (2x150bp Medium Output kit 213 samples). In total, 257 samples were utilized. Short read data can be found at `https://s3.climb.ac.uk/vapor-benchmark-data/vapor_benchmarking_realdata_reads_filtered_18_03_18.tar`. For publicly available data, any reads that were classified as human by Kraken2 (Wood and Salzberg; 2014), or those that mapped to the hg38 human genome with minimap2(Li; 2018), were removed.

These WGS datasets were then processed by extraction of influenza reads by mapping with minimap2 (Li; 2018) to 8 curated influenza segment reference fasta files (19,594 sequences in total), one at a time, produced by from all influenza segment sequences downloaded from the NIVR (`https://www.ncbi.nlm.nih.gov/genomes/FLU/`) and clustered to 99.5% identity with cd-hit-est (Li and Godzik; 2006). Extracted reads were assembled with IVA (Hunt et al.; 2015). For all 257 datasets used, a near-full length (>90%) contig could be assembled for at least one major segment protein. Samples for which a contig could not be assembled were not used. In total, 1,495 segment contigs were included.

2.3.2 Mapping assessment

Four mapping programs were assessed in this analysis: Minimap2 (Li; 2018), BWA-MEM (Li and Durbin; 2009d), NGM (Sedlazeck et al.; 2013), and Hisat2 (Kim et al.; 2015). Default settings were used for all tools. Each experiment can be reproduced using the code and instructions found at `https://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 reads from diverse samples, two simulations were performed. Firstly, for assessment of robustness to species origin, read sets were simulated with ArtificialFastqGenerator (AFG)(Frampton and Houlston; 2012) from 552 avian, 16,679 human, and 4,054 swine H1N1 HA coding sequences from the NIVR (Bao et al.; 2008a). An additional 0.05% *in silico* substitution was introduced into simulated reads to account for RT-PCR technical errors and biological intrahost variation. This rate was chosen to be in accordance with experimental observations made by Orton *et al.* (2015) (Orton et al.; 2015), although it may be conservative. Reads were then mapped to the A/California/07/2009 (H1N1) HA reference sequence. Secondly, for assessment of robustness to random 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%, which was 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, as well as vaccine recommendations. Samtools (Li et al.; 2009a) 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, 33,133 unique full-length influenza A HA coding sequences of any lineage or species were downloaded from the NIVR, and 5000 pairs 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 run with VAPOR classification, a single sequence was randomly chosen as before, but the reference was chosen by VAPOR version 1.0.1. 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 read pairs were subjected to mapping with Minimap2 with default settings for short reads (-x sr), both with and without VAPOR classification. 51 of 257 samples with less than 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), 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.

2.3.3 Algorithm overview

Definitions

Let $R = \{r_1, r_2, \cdots, r_{|R|}\}$ and $S = \{s_1, s_2, \cdots, s_{|S|}\}$ 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 $\mathcal{W} = (N, E, W)$ be a weighted De Bruijn Graph built from reads R, where N, E, W are sets of nodes (k-mers), edges ($k-1$-mer overlaps), and node weights (sequencing depth for some k-mer), for some $k \geq 2$ (by default $k = 21$). I assume a model read generation process reflective of RNA virus sequencing: let the multiset $X = \{x_1, x_2, \cdots, x_{|X|}\}$, be a population of virus sequences (quasispecies) for some gene, for which I 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). I attempt, using heuristics, to find a reference that is similar to x^*.

Mapping and Scoring

VAPOR maps each reference s against $\mathcal{W}$, such that the ith k-mer of s, denoted $s[i, i+k]$ is either mapped to some node $n \in N$, or mapped to a gap. I 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 2.1 demonstrates the concept of this mapping.

I next formulate a scoring function $f_{\mathcal{W}}(s, s')$. I chose to favor sequences for which there is high weight in $\mathcal{W}$; 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 $\mathcal{W}$ at low frequency, such that there may be several reference sequences which correspond exactly to a path in $\mathcal{W}$. Conversely, since sequencing depth in these datasets may be highly skewed, I 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, I define:

$$f_{\mathcal{W}}(s', s) = \psi(s') \cdot \sum_{i=1}^{|s|} M_i \delta(s'_i, s_i) \tag{2.1}$$

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 ith base of s'. That is, for the sequence of node weights w_i with corresponding k-mer nodes n_i of s', $M_i = \max\{w_{\bar{i}}, w_{\bar{i}+1}, \cdots, w_i\}$, where $\bar{i} = \max(0, i-k+1)$ and $\delta(s'_i, s_i) = 1$ if $s'_i = s_i$, and 0 otherwise. Since any reference can be mapped onto the graph in many ways, I attempt to heuristically find high scoring placements.

Preprocessing

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 .fastq (or .fastq.gz) file of WGS reads. Firstly, VAPOR builds a set of k-mers U from all reference sequences. Next, the ith read is decomposed into a set of non-overlapping subsequences of length k (words), A_i, and if $|A_i \cap U|/|A_i| \leq t$ (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 t, 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 $n_j \in N$ with corresponding weight $w_j \in W$ less than a coverage parameter c is discarded.

Core algorithm

Firstly, $\mathcal{W}$ 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 $\mathcal{W}$. Each reference sequence, s, with length $|s|$, 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. I seek to simultaneously perform the mapping and compute the array $M' = (M_1\delta(s_1, s'_1), M_2\delta(s_2, s'_2), \cdots, M_{|s|}\delta(s_{|s|}, s'_{|s|}))$ as in (1). Firstly, an array a is initialized from exact k-mer matches, where a_m is the weight of the mth k-mer of the reference, and any not in N are set to zero. For speed considerations, only a subset of seed arrays are extended: those with a fraction of nonzero 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 ρ 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 ith gap (run of zeros) in a of length l, a bridge b_i 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 1) inserting bridge k-mer weights and 2) re-calculating the weight at each position j as M_j (as defined in mapping and scoring). Finally, each bridge, b_i, a string, is then compared to the ith gap string, the original substring in the reference sequence corresponding to the gap, in order to compute $\delta(s'_j, s_j)$ as in (1). For any s_j in an exact

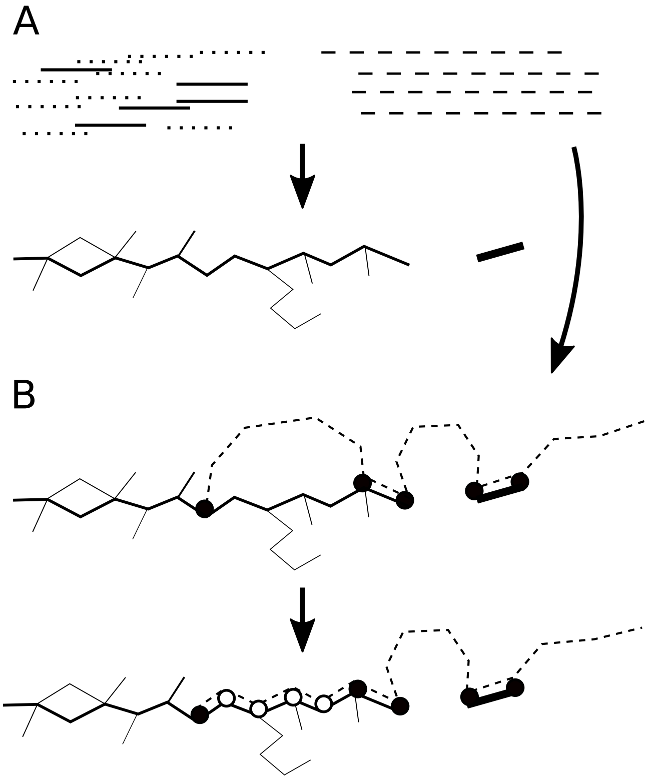


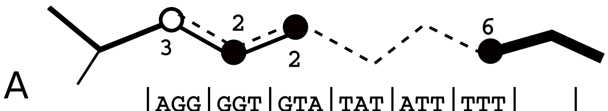
Figure 2.1: **Simplified VAPOR algorithm**. Firstly, pre-processing and graph construction is performed (A), where target reads R (solid black lines) are filtered from non-target (e.g. bacterial) reads (dotted lines) using a fast k-mer comparison to references S. This is followed by wDBG construction. Then, mapping and scoring is performed simultaneously (B), where each reference sequence s (dashed line) is mapped to the wDBG, $\mathcal{W}$, 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.

match, $\delta(s'_j, s_j) = 1$ by definition. Figure 2.2 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.

2.3.4 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_o, was chosen from 46,724 unique 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 AFG 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, I simulated an intra-host population with 4 minor sequence types, mixed in the ratio of 100:5:1:1:1, with each



		AGG TGG	GGT GGT	GTA GTA	TAT - - -	ATT - - -	TTT TTT		
B	s =	A	G	G	T	A	T	T	T
	s' =	T	G	G	T	A	T	T	T
C	a =	0	2	2	0	0	6	0	0
	a + b =	3	2	2	0	0	6	0	0
	M =	3	3	3	2	2	6	6	6
	δ(s,s') =	0	1	1	1	1	1	1	1
	M' =	0	3	3	2	2	6	6	6

Figure 2.2: **Scoring procedure for an example graph mapping.** An example mapping of a reference s (dashed line) to a graph $\mathcal{W}$ (solid lines), with exact matches (black circles) and inexact matches (white circles), is shown (A), with string representation of the path, s' (B). Firstly, a weight array a is retrieved for which a_i gives the weight of the ith reference k-mer in the wDBG, where gaps are given a weight of zero. These exact matches are then extended with bridges b to inexact matches. Next, per-base weights M are calculated such that each base is given the greatest weight of any k-mer that includes it, which also functions to assign weights to terminal characters of a string (or substring before a gap) that do not have k-mers (such as "TA" at the 4th position). Finally, the array M' is computed as $M'_i = \delta_i \cdot M_i$, where $\delta_i = 1$ if $s'_i = s_i$, and zero otherwise. For VAPOR, I chose to multiply the sum of this array by the fraction of non-gap (non-zero) positions, in order to penalize high weight, high gap mappings.

sequence additionally mutated by 1% relative to the major sequence. This process was performed 500 times for each category. Performance was assessed as follows: The Levenshtein distance of the mutated sequence s_m was taken with respect to the original sequence s_o as a baseline, denoted by $L(s_m, s_o)$; the reads were classified by each tool with all 32,804 references as a database, and the best hit s_c returned by each were compared to the mutated sequence to obtain $L(s_m, s_c)$. Global alignment was performed with the pairwise2 module of Biopython(Cock et al.; 2009) (with cost parameters 0, -1, -1, -1). I defined the additional Levenshtein distance, $L_A = L(s_m, s_c) - L(s_m, s_o)$. This distance was chosen because, for mutated sequences, it captures the additional error in classification beyond that caused by uniform mutation to the original reference. I note that $L(s_o, s_m)$ may occasionally be sub-optimal, that is there may exist s'_o such that $L(s'_o, s_m) < L(s_o, s_m)$ where *in silico* mutations introduced resulted in a sequence more similar to some other sequence in the database than the original.

For real datasets, 257 raw read sets that produced full-length contigs for at least one segment were chosen from the sequencing runs described above. The assembled contigs were annotated with BLAST (sorting by e-value, bit-score, and length), and raw reads classified by VAPOR. The percentage identity of VAPOR classifications to each contig was recorded.

2.3.5 Detection of reassortments and zoonotic strains

For assessment of reassortment classification, two simulations were performed. Firstly, 9659 avian, 18,308 human, and 2893 swine complete influenza genome sets were downloaded from the NIVR. 250 human genome sets were randomly selected. Another 250 were randomly selected with a single segment swapped with a randomly chosen avian or swine segment. For each, 1000 reads from each segment were simulated uniformly with an error rate of 0.5%. Each set of reads was classified with VAPOR. For the reference strains chosen by VAPOR for each segment, respective HA sequences were compared by global alignment, and percentage identity (PID) taken. If the maximum pairwise distance between chosen strain HA sequences exceeded a given threshold v, a classification of true was returned. Receiver operating characteristic (ROC) curves were generated by varying the parameter v. For assessment of intra-subtype reassortment classification, the same experiment was performed with randomly chosen H3N2 genomes.

2.3.6 Computational resources

In all cases, experiments were performed natively on a 96 core, 1.4 TB memory CentOS version 7.4.1708 virtual machine hosted by CLIMB (Connor et al.; 2016), with GNU parallel (Tange; 2011) where required.

2.4 Results

2.4.1 Benchmarking single-reference mapping

A range of mapping programs (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 NIVR 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 the real datasets, which was frequently higher than 2%. The proportion of successfully mapped reads for each tool and host species is given in Figure 2.3. In this case, using a single reference strain with any of the programs 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 uncharacterized. I note that this analysis is not an evaluation of overall mapping performance, since such an analysis must include mapping scores, but evidence that regardless of software, data loss may potentially occur.

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* per-base uniform mutation with a specified probability, with additional read error of 0.05% as before. These results, shown in Figure 2.4, demonstrate that, for all mapping programs, at approximately 10% mutation, read recovery begins to regularly diminish, which is insufficient for robust mapping of influenza strains from different species given high diversity and mutation rate. Furthermore, for several of the programs tested, mapping quality was suboptimal beyond 1-3% mutation.

2.4.2 Classification performance simulation

Since utilizing a single, standard reference can have complications, I explored methods for reference selection. In order to assess the performance of classification from simulated reads, VAPOR was compared to MASH and consensus BLAST classification. Reads were simulated from randomly selected NIVR 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 2.5 shows the additional Levenshtein distance, $L_A = L(s_m, s_c) - L(s_m, s_o)$, for each tool, where s_o, s_m, and s_c are the original, mutated, and retrieved database sequences respectively. 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

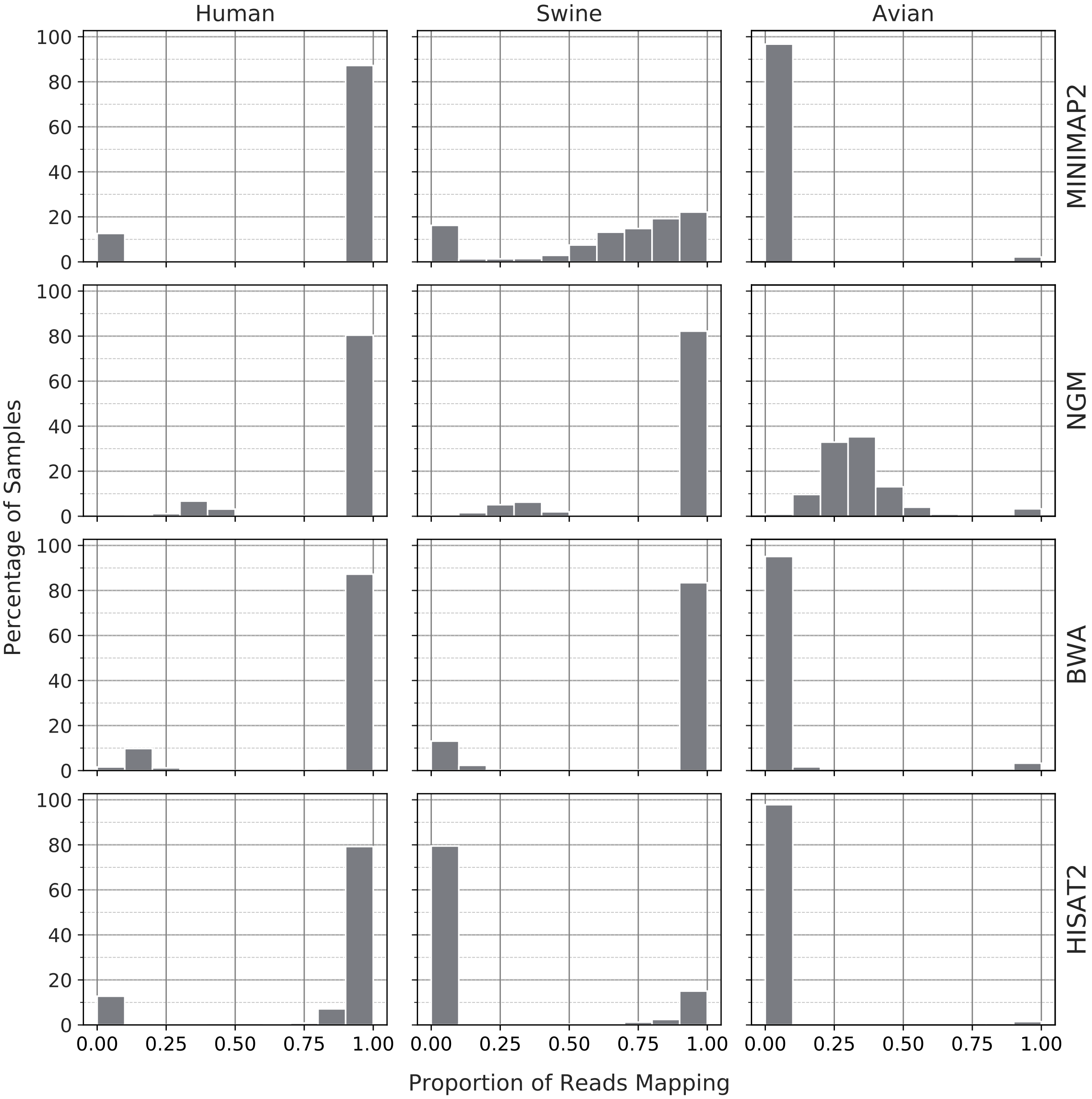


Figure 2.3: **Density histograms showing proportion of mapped reads in samples, by software and dataset.** Reads were simulated for each dataset retrieved from the NIVR: 16,679 Human H1N1 HA (left column); 552 avian H1N1 HA (middle column); 4054 Swine H1N1 HA (right column). All sequences were mapped to California/07/2009. For human sequences, most simulated datasets mapped successfully, although even for this dataset, around 10% of samples had some proportion of unmapped reads. However, for avian and swine sequences, mapping quality was poor, and often failed entirely. Even for the best performing software, NGM, avian sequences in particular mapped poorly.

and VAPOR were often near-optimal or optimal, despite a large amount of noise, and that the performance difference between these approaches was very small. These results show that the algorithm used by VAPOR facilitates accurate classification of influenza strains directly from reads, comparable in accuracy but faster than BLAST for WGS read sets, which is generally not computationally tractable for datasets with millions of reads.

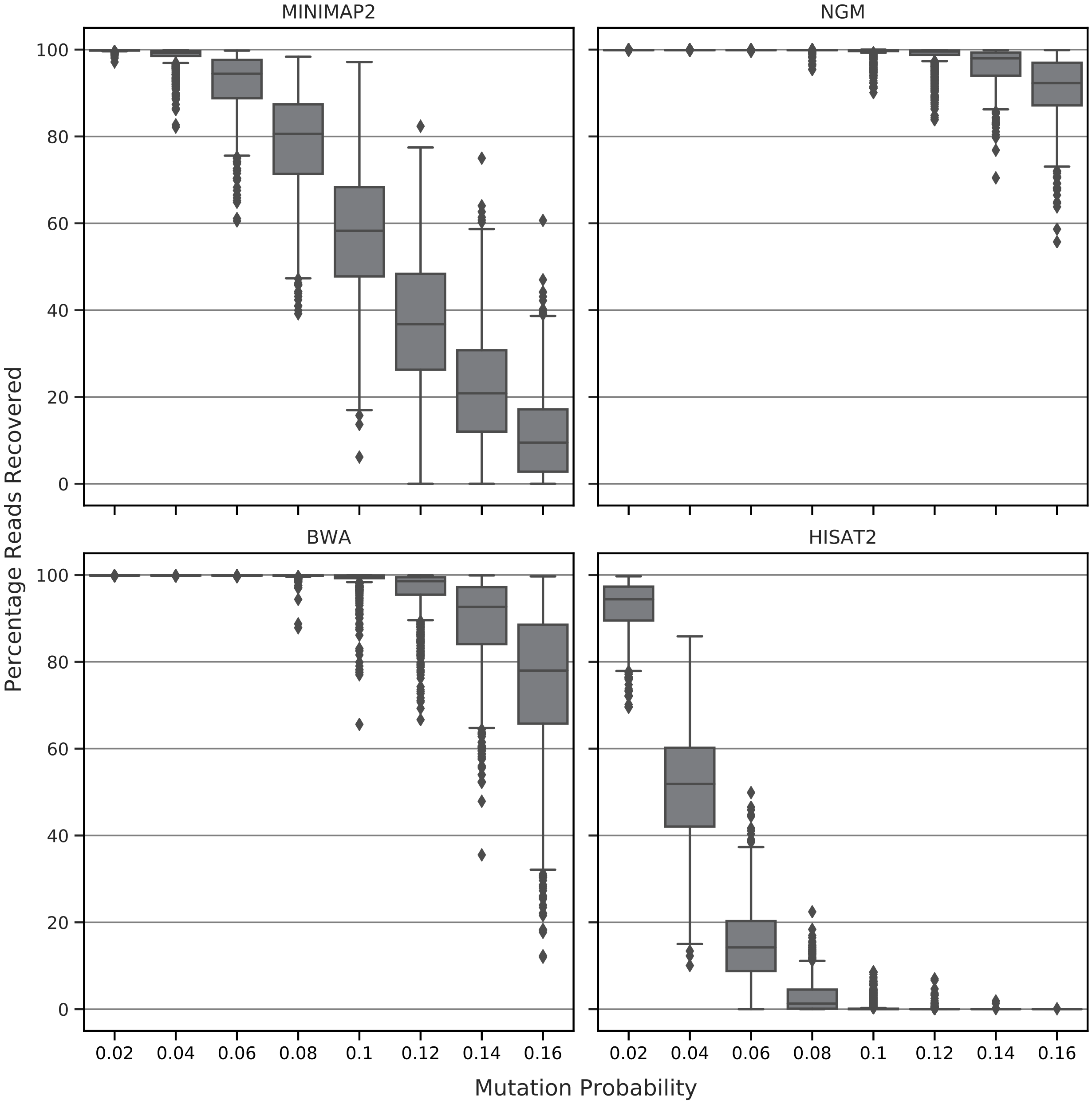


Figure 2.4: **Box plots showing percentage of simulated Human H3N2 HA reads mapping to Perth/16/2009 for each software, with *in silico* uniform mutation at indicated per-base probability**. Reads were simulated from *in silico* uniformly mutated Perth/16/2009 HA with the indicated per-base probability, approximately corresponding to 2 to 16% divergence. Reads were additionally subjected to 0.05% substitution to account for technical noise, such as from RT-PCR, and biological noise, such as from intrahost variation. Data loss was frequently observed with all tools beyond 10% mutation. Outliers are indicated as diamonds. N=1000 for each category.

2.4.3 Real data classification performance

Unlike BLAST and MASH, VAPOR can be applied directly on reads with no pre-processing. As such, in order to validate the performance of VAPOR directly on real datasets, I 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

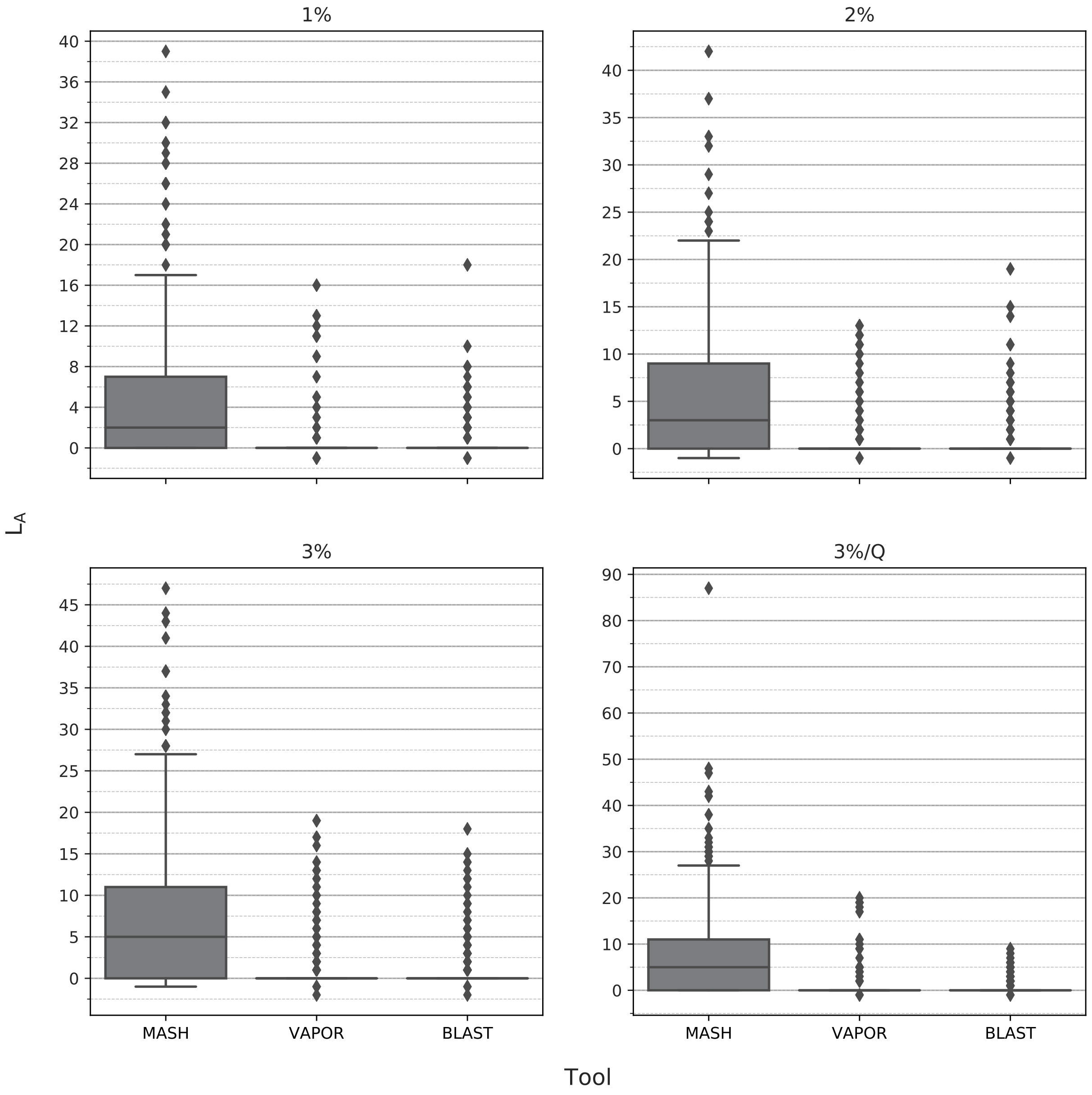


Figure 2.5: **Box plots showing additional Levenshtein distance $L_A = L(s_m, s_c) - L(s_m, s_o)$ of input sequence to output reference chosen by VAPOR, MASH, and BLAST consensus classification.** Reads with 1% error rate were generated from randomly selected references mutated *in silico* by 1%, 2%, 3% and 3% with additional biological intra-host variant noise simulation 3%/Q, and repeated 500 times for each category. L_A is defined as Levenshtein distance of a classified sequence s_c to original mutated sequence s_m, minus the distance of the original mutated sequence s_m to the original non-mutated reference sequence s_o. Outliers are indicated as diamonds. Performance of VAPOR was generally equivalent to that of BLAST. For both of these tools, classification most often resulted in none, or a few extra incorrect bases. Sequences ranked highest by MASH were often sub-optimal.

reads were classified by VAPOR. The chosen reference was compared by global alignment to the assembled full length contigs. Figure 2.6 gives a scatter plot showing the PID of references retrieved by VAPOR 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

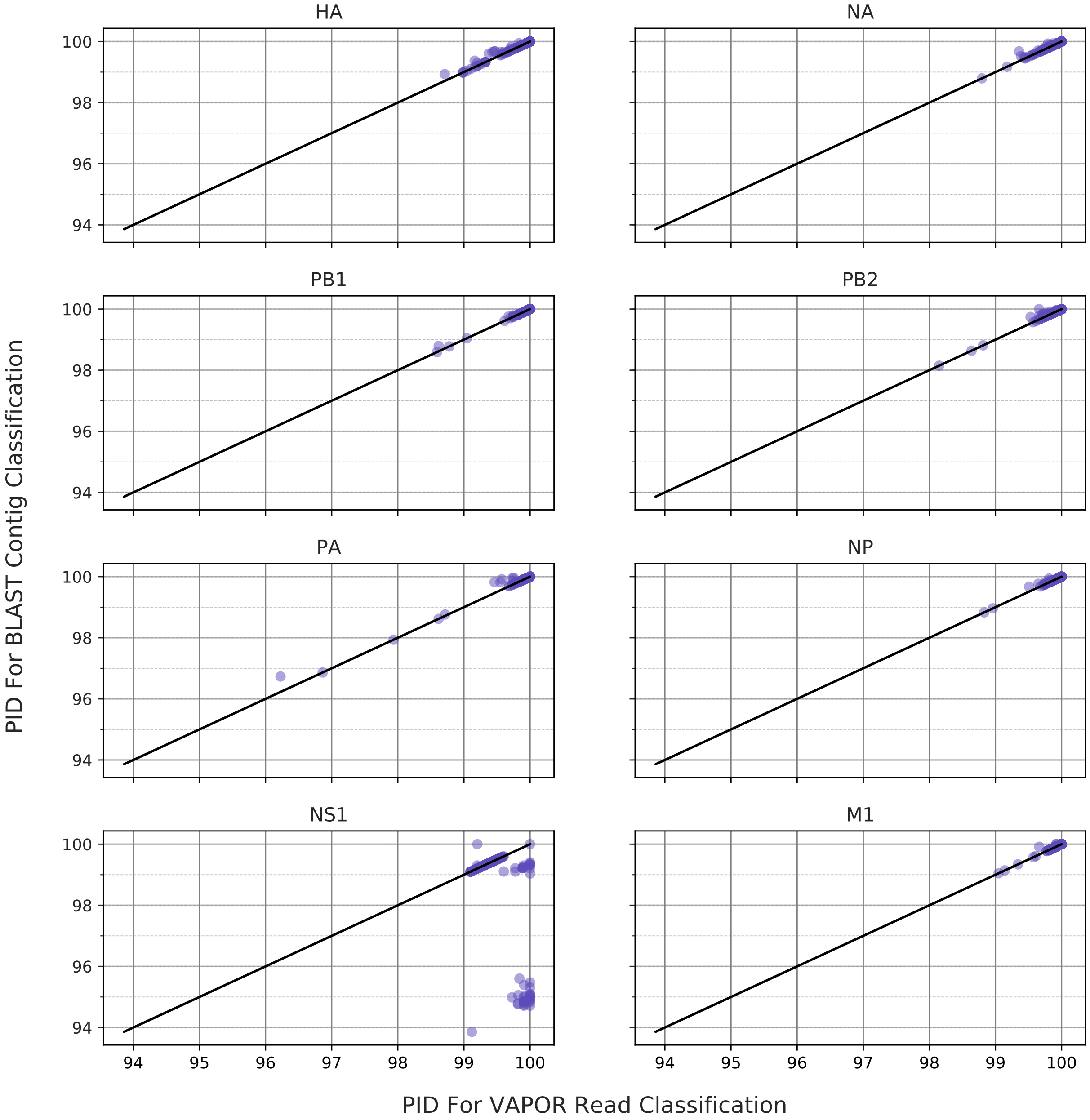


Figure 2.6: **Scatterplots showing PIDs of VAPOR read classifications versus BLAST contig classifications with respect to assembled contigs for all 8 major segment coding sequences**. Black lines indicate $x = y$. Points that fall below this line were classified better from reads with VAPOR. Points above the line were classified better with BLAST from contigs. VAPOR is capable in general of performing classification of reads to within 1% of the correct sequence. The mean PID of VAPOR classifications for all segments was 99.82%. For datapoints under 98% PID, BLAST was generally also not able of providing a better classification given the reference database.

classification. The mean percentage identity 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. 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. I note that, for some contigs, neither BLAST nor VAPOR could achieve classifications with a PID greater

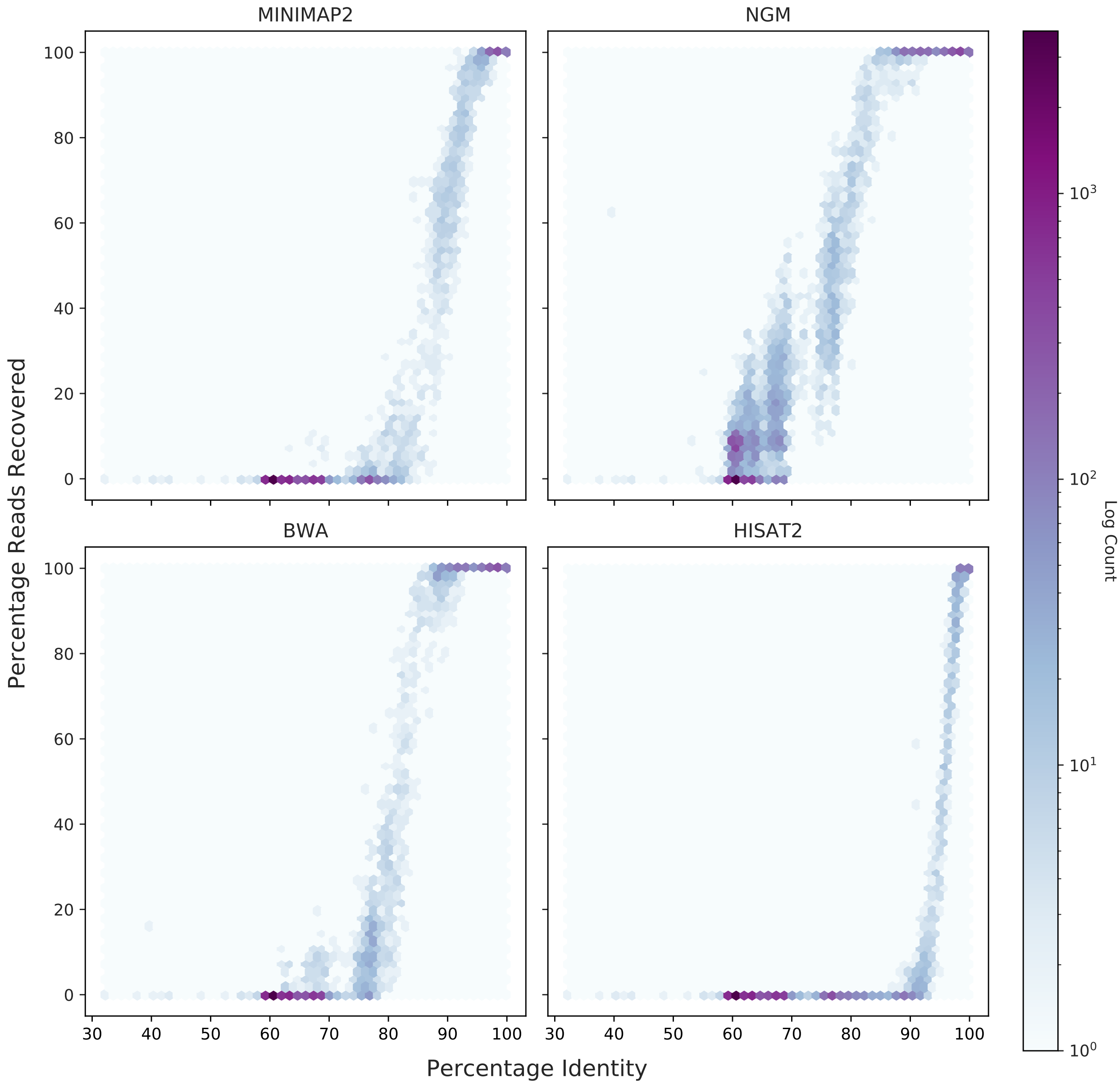


Figure 2.7: **Read recovery versus percentage identity for randomly chosen influenza A HA sequence pairs**. Influenza A HA sequences were chosen randomly from the NIVR database in pairs, one used as a reference, and one used to simulate reads for mapping. For percentage identities lower than 10%, mapping was unreliable. Furthermore, notably between strains of differing host origin, percentage identity was observed in many cases to be as low as 60%. These trends were observed for all tools, although performance dropped off more gradually for NGM.

than 97%. Manual examination of these samples showed large deletions, with at least one a likely misassembly (deletion including start codon, inclusion of 5' UTR).

2.4.4 Mapping with pre-classification

In order to assess the utility of pre-classification for mapping, 25,533 full-length HA coding sequences from human, avian, and swine hosts were downloaded from the NIVR, and pairs were chosen ran-

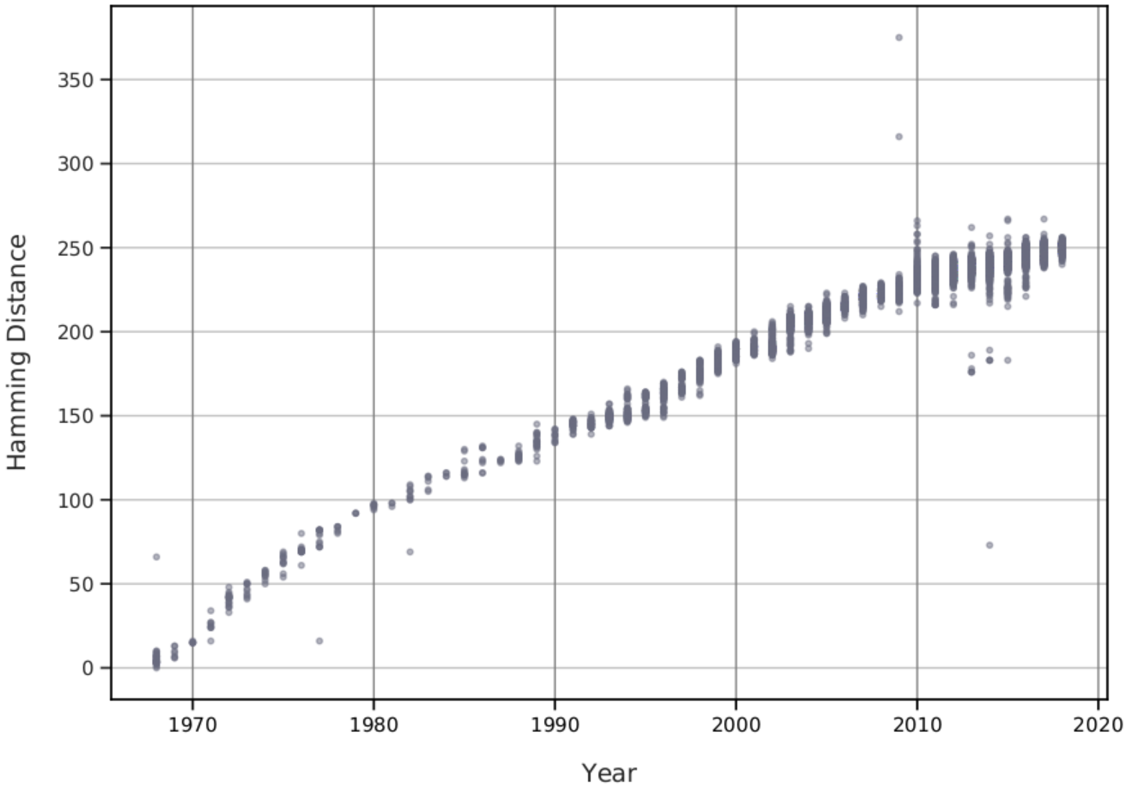


Figure 2.8: **Hamming distance of public database sequences from A/Perth/16/2009..**

domly; one was used for read simulation, and the other as a mapping reference. In this case, mapping was performed both with and without pre-classification with VAPOR. Figure 2.7 gives the percentage of reads recovered against percentage identity for a pair, which shows that, for pairs chosen with less than 90% percentage identity, read recovery was poor. For mapping without pre-classification, mean recovery rates for Minimap2, NGM, BWA-MEM, and Hisat2 were 12.4%, 23.1%, 15.9%, 6.9%. However, with pre-classification using VAPOR, the mean was over 99.72% for all tools. These results demonstrate that mapping pipelines that include pre-classification are robust to sequences of zoonotic origin. Raw mapping performance was also assessed on real data by mapping datasets with Minimap2 with and without pre-classification. Figure 2.9 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 6.85%, 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 may represent a true

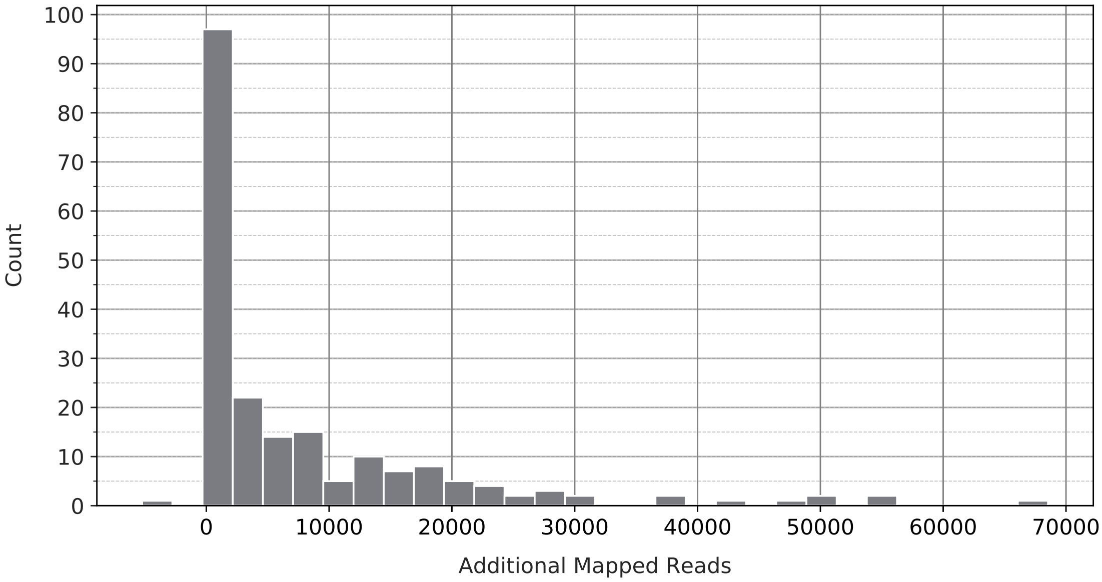


Figure 2.9: **Additional Number of reads mapped by Minimap2 with VAPOR pre-classification for 257 real WGS datasets.** Pre-classification with VAPOR on average resulted in 7816.03 more mapped reads. Several samples gained more than 50,000 reads by choosing a suitable reference. For one sample, representing a possible coinfection, 5221 fewer reads mapped when using a single reference chosen by VAPOR.

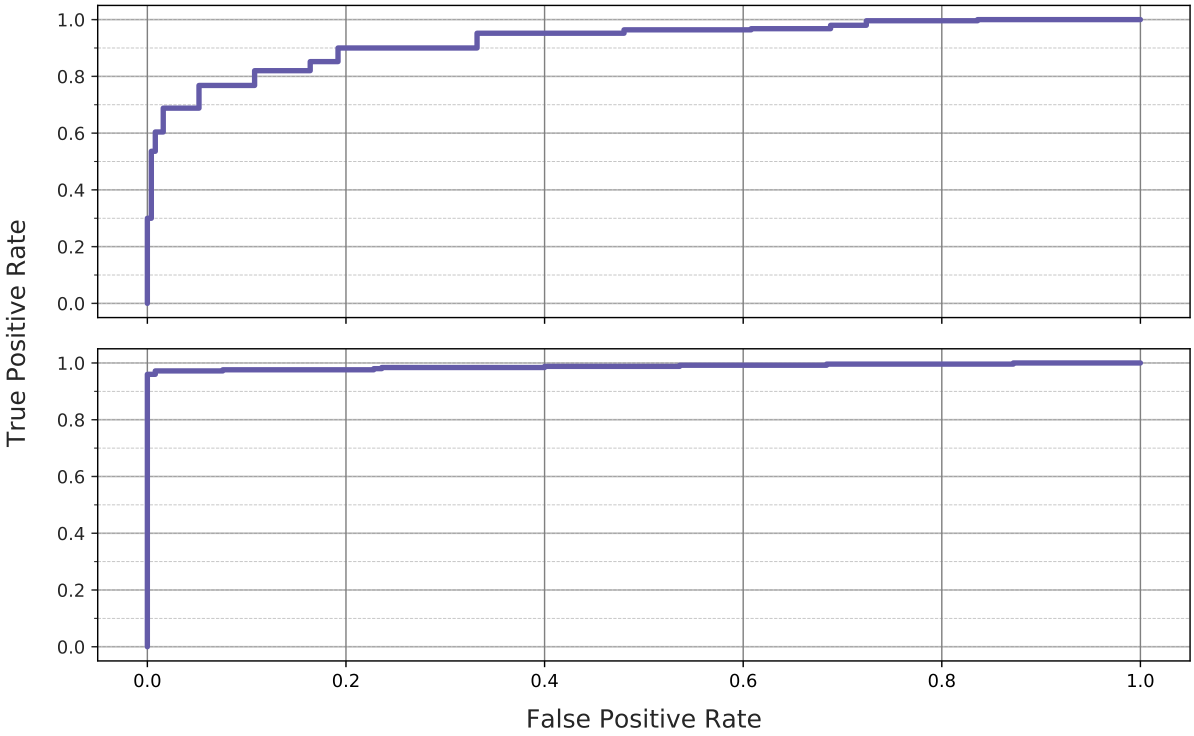


Figure 2.10: **Receiver operating characteristic (ROC) curve for classification of simulated reassortment events.** Although most zoonotic reassortments were detected at all parameter values (top), intra-H3N2 lineage (bottom) reassortments were more difficult to detect. The curve was generated by varying v, the minimum PID between VAPOR classifications of individual segment strains on the basis of HA. Due to the noise present in VAPOR classifications, as well as the close sequence similarity of H3N2 sequences, all parameter values with high TPR corresponded to a large FPR.

biological coinfection or a contamination, and could not be mapped to a single reference.

2.4.5 Detection of reassorted strains directly from reads

In order to assess the application of read pre-classification to reassortment detection directly from reads, 250 simulated reassortment events with zoonotic strains were mixed with 250 complete genome sets, reads simulated, then classified by VAPOR. A simple reassortment classifier was used on the output of VAPOR, which compared the minimum pairwise PID of the HA sequences of the 8 strains assigned by VAPOR to each segment; if this PID was below a given parameter v, a reassortment was called. A ROC curve is shown in Figure 2.10, illustrating the performance of this classification strategy. Simulated zoonotic reassortments were detected with 97.2% true positive rate (TPR) and 0.08% false positive rate (FPR) for a 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 less than 90% to human strains, depending on origin. I 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. I found at a PID threshold of 96.3%, a TPR of 76.8% could be achieved at a FPR 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.

2.5 Discussion

2.5.1 Mapping approaches and improvement with VAPOR

I provide evidence that, in the best case, approaches for influenza virus analysis that use mapping to a single reference may result in small amounts of data loss due to biological variation and noise. This problem has potential to worsen over time given continued and rapid divergence, and could have implications for standard references described by organizations such as the WHO. However, in the worst case, using a hard-coded reference sequence in bioinformatics pipelines can result in data loss. As shown in Figure 2.8, influenza strains continually accumulate substitutions relative to a single reference (approximately 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 the worst cases, mapping may fail completely, when usable data is present, requiring time and expertise to resolve with more complex methods. Our approach largely avoids these pitfalls altogether, allowing much simpler pipelines and alignment visualization via standard genome browsers, while also retaining the advantages of using a mapping-based approach for analysis. I chose Minimap2, BWA-MEM, NGM, and Hisat2 in order to represent a range of mapping softwares. BWA in particular has found use in general for influenza read mapping (Rutvisuttinunt et al.; 2013)(Borges et al.; 2018)(Yu et al.; 2014)(Wu et al.; 2014)(Imai et al.; 2018a)(Leonard et al.; 2016)(Jonges et al.; 2014). In other cases, software such as Bowtie2(Langmead and Salzberg; 2012) have been used (Meinel et al.; 2018a)(Goldstein et al.; 2017) for mapping to single references. 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 programs, 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). In other cases, different single references have been used (Meinel et al.; 2018a). They represent single strains that are well known, and may be used to represent each subtype. I do not believe that using different individual strains would affect the trends demonstrated.

I note that alternative approaches exist, including mapping to a large sequence database, but this does not allow for visualization of an alignment, and subsequent analysis such as characterization of point mutations. I note that in principle, pre-classification with any software could work reasonably well. MASH performed well in simulations. However; using an 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. With regards to *de novo* assembly, in the cases where not enough initial data exists to assemble fragments, mapping allows analysis of limited fragments. 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).

In all but one of the real data cases examined, pre-classification with VAPOR resulted in a greater number of mapped reads than mapping to 4 reference strains from 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 coinfection detection. In these relatively rare cases, a single reference is not sufficient for mapping.

2.5.2 VAPOR algorithm and performance

I have presented a novel approach to virus classification from short reads data using DBGs. However, studies on the use of graph data structures in bioinformatics have been accumulating. These studies could be leveraged to improve the speed and memory efficiency of VAPOR, via efficient DBG representations, or graph traversal algorithms. In future study, as public sequence data accumulates, `VAPOR` may show promise in WGS approaches for other RNA viruses with small genomes, such as measles virus, Hepatitis C Virus, Human Immunodeficiency Virus (HIV), or Ebola virus. In general, this approach may have applications to short, variable genomes with high redundancy databases. I have shown that in many cases, VAPOR outperforms MASH, and has comparable performance to BLAST-based approaches. Furthermore, the algorithm used by VAPOR is well suited to simultaneous pre-filtering of contaminating human or bacterial sequences in samples, although I note that, in cases of coinfection, the developed algorithm may not be sufficient. Lastly, improved speed may be achieved by future implementation in C++, although generally, for the datasets examined, VAPOR can run within 5 minutes on a laptop with a 2.60GHz i7-6600U CPU.

Several default parameters were explored during development, but not exhaustively. A k-mer 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 kmers 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.

2.5.3 Real-data classification

As demonstrated in 2.6, the BLAST contig classification strategy I used performed poorly on NS1. This was due to sorting by e-value, bit-score, and length over percentage identity, combined with the presence of some NS1 sequences in the database which were longer than the required coding region. I opted to include this result to illustrate a potential pitfall that can occur with automated BLAST classification. 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. Furthermore, samples with deletions of ambiguous origin could not be excluded.

2.5.4 Reassortment classification

Over 97% of simulated zoonotic transmission events or reassortments could be identified at a cost of a 0.08% FPR using a simple alignment strategy whereby the PID of the HA sequences corresponding to the strains assigned to each segment are compared. Whilst some false positives occurred, this strategy provides a basis for pre-screening that can then be confirmed with slower methods as required. Intra-subtype classification from a single host species, such as human H3N2 was more difficult to classify. In this case, reported positives could be further validated by slower methods such as phylogenetic placement of assembled contigs. I note that it is not known *a priori* if any of the NIVR genomes are reassortments themselves. It is also possible that randomly choosing zoonotic strains to reassort is not biologically accurate, since there may be a limit on the similarity of reassorted sequences. However, I applied the same methodology to H3N2 sequences in order to demonstrate feasibility in detecting reassortment between very similar strains, although this was less accurate.

2.5.5 Conclusions

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

2.6 Availability of data and materials

All scripts and pipelines used for simulations can be found `https://github.com/connor-lab/` in the following repositories: `vapor_benchmark_mapping`; `vapor_benchmark_simulation`; `vapor_benchmark_realdata`; `vapor_benchmark_simulation`. Short read data can be found at `https://s3.climb.ac.uk/vapor-benchmark-data/vapor_benchmarking_realdata_reads_filtered_18_03_18.tar`. Human sequences were depleted from this data as described in Methodology. All other data required for reproduction of results can be obtained according to the instructions found in the respective repositories.

2.7 Acknowledgments

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

In addition, I would like to thank Thomas Connor and Matthew Bull for intellectual input and contextual discussions of previous pipeline implementations, and Catherine Moore and Thomas Connor for orchestrating sequencing initiatives at PHW. I would like to thank Ben Southgate for additional help with software implementation and testing.

Chapter 3

Influenza virus coinfection and contamination detection for automated WGS pipelines

3.1 Abstract

In order for a RNA virus bioinformatics pipeline to be robust, it must be able to handle edge cases. Here, I examine approaches to detection of mixed samples in order for them to be properly handled and analyzed. True influenza coinfections, whilst rare, are both clinically relevant, and allow reassortment to occur, which is the mechanism by which pandemic strains may arise. Furthermore, contamination between samples, which is expected to happen, is important from a quality control standpoint, since these samples may result in data loss or sequence bias. In the context of routine, high-throughput bioinformatics pipelines, it is therefore important that mixed samples are reliably detected and quantified, and likely contaminations differentiated from coinfections. Existing bioinformatics methods for this task are either based on broad sequence classification, which can perform well for divergent sequences, or designed for quasispecies quantification. As such, I developed a method based on mixture modelling and expectation-maximization, commonly used for full quasispecies estimation, for the purpose of quantifying and classifying contamination between closely related samples in the context of a whole-genome sequencing pipeline. These results, using simulations and mixtures of real data, indicate that mixtures can be reliably detected, even between very closely related sequences. For simulations, I found the average error in estimated mixture proportion to be less than 2%. Furthermore, I identified 10 real data samples as candidate mixtures, with 4 of these mixtures with a mixing proportion greater than 10%.

3.2 Introduction

During the process of sequencing, what is believed to be a sample representing a single virus genome may in fact be a mixture resulting from coinfection or contamination. The former may have important clinical (such as for patient outcome), epidemiological or evolutionary (such as resultant reassortment or recombination) consequences. The latter may complicate the sequence assembly process or result in the generation of erroneous sequences, which can be found in relatively large quantities in on-line databases (Rayko and Komissarov; 2020).

3.2.1 Influenza virus coinfection

Coinfection of individuals with influenza A and B has previously been observed (Perez-Garcia et al.; 2016), as well as between viruses of mixed subtype, such as between A/H3N2 and A/H1N1 (Falchi et al.; 2008; Kendal et al.; 1979; Lee et al.; 2010; Ducatez et al.; 2010; Liu et al.; 2010; Myers et al.; 2011), or between seasonal H1N1 and H1N1pdm09 (Ducatez et al.; 2010). For example, Rith *et al.* (2015) recorded a coinfection between A/H3N2 and A/H1N1pdm09 that resulted in a reassortant virus (Rith et al.; 2015), which led to their suggestion that coinfections should be considered during routine surveillance. Proportions of coinfecting viruses have been reportedly such that one strain is present in a dominant proportion (Falchi et al.; 2008), but (Rith et al.; 2015) reported high viral titers for both. It is well known that coinfection with different influenza viruses can result in pandemic strains. Indeed, this is believed to be the direct mechanism whereby they arise (de Silva, Tanaka, Nakamura, Goto and Yasunaga; 2012). One potential implication of coinfection between seasonal viruses and pandemic strains is that a further reassortment will result in enhanced pathogenicity or antiviral resistance (Lee et al.; 2010; Schrauwen et al.; 2011; Peacey et al.; 2010). In support of this concern, coinfections between the H1N1 pandemic strain and seasonal H1N1 were observed (Peacey et al.; 2010). Even between currently circulating strains, coinfection can result in reassortants, with examples such as H1N2 observed from reassortment between A/H3N2 and A/H1N1pdm09 (Wiman et al.; 2019). Althogh coinfections can be confirmed with specific PCR, depending on coinfecting viruses (Peacey et al.; 2010), currently the detection of putative mixed infections in bioinformatics pipelines is performed by heuristic procedures or not at all (Wan et al.; 2015; Zheng et al.; 2017; Borges et al.; 2018; McGinnis et al.; 2016b), unless the mixture is composed of viruses from disparate subtypes, in which case standard taxonomic identification can be performed. For example, INSaFLU relies on flagging samples with a large number of SNVs. However, the reliability of such a procedure has not been established.

3.2.2 Experimental objectives

Given that an extensive body of literature exists for quantification of complex quasispecies populations, I believed that bioinformatics pipelines could incorporate more specific methods for detecting mixed samples than heuristics based on the number of SNVs. As such, in this study, my primary objectives were:

1. Formulation of a suitable mixture model for co-infections and contaminations.
2. Development of two methods of inference based on EM: i) one allowing the unconstrained estimation of a coinfecting sequence from an alignment and ii) one where the alternative coinfecting sequence is constrained to a panel of references.
3. Development of an automated decision rule for determining whether a dataset is a coinfection. In particular, I aimed to assess the modified LRT for mixture models under certain conditions.
4. In principle, mixture modelling and EM are routine tasks for more conventional data-types, with existing libraries. However, I aimed to perform mixture modelling with a full read alignments as data, instead of reducing the data to multinomial counts at sites. As such, I aimed to implement our method as software with improvements to allow fast likelihood calculations.

3.3 Methodology

3.3.1 Mixture model formulation

Let a collection of n reads, denoted X, be generated by a mixture of two virus genomes s_0 and s_1 with proportions π and $1 - \pi$, respectively, and a common error rate γ. The probability of observing read i is modeled as:

$$P(X_i = x|s_0, s_1, \pi, \gamma) = \pi P_0(X_i = x|s_0, \gamma) + (1 - \pi)P_1(X_i = x|s_1, \gamma)$$

Here, the generation probability mass functions P_0 and P_1 are assumed to be of the form:

$$P_k(X_i = x|s_k, \gamma) = \prod_{j=1}^{l} (1 - \gamma)\delta_{s_{k,j}}(X_{i,j}) + \gamma(1 - \delta_{s_{k,j}}(X_{i,j}))$$

where $\delta_{s_{k,j}}$ is the Kronecker delta function, which is 1 if $s_{k,j} = X_{ij}$ (utilizing a global index for j), and zero otherwise. For position- and base-independent (no base quality scores) error rates, this is $(1 - \gamma)^{N_{match}} \gamma^{N_{mismatch}}$. I justify this simplification of population structure by pointing to previous observations that, at least for influenza, *de novo* minor variants are uncommon and in low frequency. The number of variants across the entire genome with greater than one percent or so frequency is typically less than 15, with a proportion usually less than 10%. (Debbink et al.; 2017; Xue et al.; 2018). Therefore, I define a model of coinfection, as opposed to intra-host diversity, as two strains of some distance d, and with some appreciable proportions π and $1-\pi$. Here, the parameter γ captures biological population variation, assumed to arise by mutation, as well as RT-PCR errors, and errors generated from reads. For computational purposes, I make the assumption that s_0 is fixed as the consensus, in which case the model resembles an admixture (Di and Liang; 2011).

3.3.2 Real datasets

Of 257 WGS previously published (Southgate et al.; 2020) short read sequencing `.fastq` files (see Chapter 3), 173 assembled to produce full length HA sequences, and 138 of these were influenza A H3N2 or H1N1. Samples that did not produce assemblies, or those from influenza B, were not used. For simulated mixtures of real data, forward reads were mapped with minimap2 (Li; 2018) to sample 100, and sam files were processed with samtools (Li et al.; 2009a). 190 pairs of alignments were then chosen and mixed together. Since, for many real datasets, coverage is extremely skewed, a subsampling strategy across the genome was utilized, where reads were sequentially sampled randomly at each position from left to right, repeatedly.

3.3.3 Dataset simulation

A reference was selected for s_0 at random from a reference set $\mathcal{R}$ of 8942 known full length influenza A HA sequences from the NCBI influenza virus resource (Bao et al.; 2008b), with s_1 selected after from references differing by at least one position, then mutated randomly at two more positions each to model sequence novelty. Additionally, in order to simulate more realistic populations, additional minor quasispecies were generated by a stick-breaking process with parameter $\alpha = 5$. For the stick-breaking process, 90% of the mass was assigned to the input (in order to increase the concentration), and the remaining 10% distributed by $p_k = V_k \prod_{k'=1}^{k-1}(1 - V_{k'})$ with $V_k \sim \mathrm{Beta}(1, 5)$ (Ren et al.; 2011) (the maximum frequency was chosen to be the input). FASTQ files with `--nreads` reads, with each read of length `--read_length` 200, were then simulated from these populations by uniform mutation with per-base probability γ; the population represented by s_0 was sampled with probability π, and that of s_1 with probability $(1 - \pi)$. Reads covering unknown bases in the reference were excluded, incorporating basic sample coverage variation. These simulations were performed with unpaired reads. Finally, simulated reads were mapped to the major consensus sequence with minimap2 (Li; 2018), and bams were indexed with samtools (Li et al.; 2009a). For estimation, .bam files were imported with pysam, a python wrapper for htslib (found at `https://github.com/pysam-developers/pysam`).

3.3.4 Expectation-maximization

Expectation-Maximization (EM) was used as described previously (Dempster et al.; 1977). For each read X_i, Let Z_i be a latent class variable, indicating whether the reads was generated from genome 0 or genome 1. For brevity let $\delta_{s_{k,j}}(X_{ij})$ be denoted $\delta_{s_k,j}$.

$$P(Z_i = k | X_i = x, \gamma, s_0) \propto P(X_i = x | Z_i = k, \gamma, s_0) = \prod_j \Big((1-\gamma)\delta_{s_k,j} + \gamma(1 - \delta_{s_k,j}) \Big)$$

As previously described for mixture models (Fan et al.; 2010), we compute the array T for each iteration (where θ denotes the set of model parameters).

$$T_{i,k}^{(t)} = P(Z_i = k | X_i = x_i; \theta^{(t)}) = \frac{\pi_k^{(t)} P(X_i = x_i | Z_i = k, \theta^{(t)})}{P(X_i = x_i | \theta^{(t)})}$$

This array, at each iteration t, gives the membership probabilities for each read given the parameters at t. Next, I utilize the Q function:

$$\begin{aligned} Q(\theta|\theta^{(t)}) &= \sum_{i=1}^{n} E_{Z_i|X_i,\theta^{(t)}} \Big[\log L(\theta; X_i, Z_i) \Big] \\ &= \sum_{i=1}^{n} \sum_{k=0}^{1} T_{i,k}^{(t)} \log P(X_i = x | \gamma, s_0, s_1, Z_i = k) + \sum_{i=1}^{n} \sum_{k=0}^{1} T_{i,k}^{(t)} \log \pi_k \end{aligned}$$

I next seek to maximize this function. Note that the above sum can be separated into two parts which are maximized independently. From standard theory (Fan et al.; 2010) it can be shown that the π_k that maximizes this function is given by:

$$\pi_k^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} T_{i,k}^{(t)}$$

Next, I consider the other parameters s_0, s_1, γ. Let j be the global (genome) index of the jth base of the ith read.

$$\begin{aligned} (\gamma^{(t+1)}, s_0^{(t+1)}, s_1^{(t+1)}) = \arg\max_{\gamma, s_0, s_1} \sum_{i=1}^{n} \sum_{j} T_{i,0}^{(t)} \log \Big(\delta_{s_0,j}(1-\gamma) + (1 - \delta_{s_0,j})\gamma \Big) \\ + T_{i,1}^{(t)} \log \Big(\delta_{s_1,j}(1-\gamma) + (1 - \delta_{s_1,j})\gamma \Big) \end{aligned}$$

Since, for any genome cluster s_k, and read base j,

$$\log\Big(\delta_{s_k,j}(1-\gamma)+(1-\delta_{s_k,j})\gamma\Big)=\delta_{s_k,j}\log(1-\gamma)+(1-\delta_{s_k,j})\log(\gamma)$$

However, instead of iterating over read indices j, we can re-write the function to be maximized in terms of matching and mismatching genome indices (where J_0 and J_0' are sets of matching and mismatching positions, respectively), which I denote W:

$$W(\gamma,s_0,s_1|\theta^{(t)})=\sum_{i=1}^{n}\Big[\sum_{j\in J_0}T_{i,0}^{(t)}\log(1-\gamma)+\sum_{j\in J_0'}T_{i,0}^{(t)}\log\gamma+\sum_{j\in J_1}T_{i,1}^{(t)}\log(1-\gamma)+\sum_{j\in J_1'}T_{i,1}\log\gamma\Big]$$

Or, we can index the same sum by iterating over genome indices first, and reads that map to those indices. Let V_j be the set of reads that covers position j, then:

$$\begin{aligned}W(\gamma,s_0,s_1|\theta^{(t)})&=\sum_{i=1}^{n}\Big[\sum_{j\in J_0}T_{i,0}^{(t)}\log(1-\gamma)+\sum_{j\in J_0'}T_{i,0}^{(t)}\log\gamma+\sum_{j\in J_1}T_{i,1}^{(t)}\log(1-\gamma)+\sum_{j\in J_1'}T_{i,1}\log\gamma\Big]\\&=\sum_{j=1}^{L}\sum_{i\in V_j}\log(1-\gamma)T_{i,0}^{(t)}\delta_{s_0,j}+\log\gamma T_{i,0}^{(t)}(1-\delta_{s_0,j})\\&\quad+\sum_{j=1}^{L}\sum_{i\in V_j}\log(1-\gamma)T_{i,1}^{(t)}\delta_{s_1,j}+\log\gamma T_{i,1}^{(t)}(1-\delta_{s_1,j})\\&=\sum_{j=1}^{L}W_{j0}(s_{0,j},\gamma)+\sum_{j=1}^{L}W_{j1}(s_{1,j},\gamma)\end{aligned}$$

This leads us to the following:

Theorem 3.3.4.1 *For all $\gamma\in(0,0.5)$, $W(\gamma,s_0,s_1|\theta^{(t)})$ is maximized by the strings s_0^*,s_1^* for which position $s_{0,j}^*,s_{1,j}^*$ minimizes the total number of mismatches, weighted by membership probability of reads covering position j. That is:*

$$s_{0,j}^*=\arg\max_{c\in\Sigma}\sum_{i=1}^{n}\delta(X_{i,j},c)T_{i,0}$$

and

$$s_{1,j}^*=\arg\max_{c\in\Sigma}\sum_{i=1}^{n}\delta(X_{i,j},c)T_{i,1}$$

independently of the value of γ (provided it is stricty less than 0.5).

Proof:

1. Let γ be fixed in $(0, 0.5)$. Let V_j be the set of reads that covers position j. Importantly, we can index the sums by genome position. Let the quantity we seek to maximize, conditional on γ, be $W(\gamma, s_0, s_1|\theta^{(t)})$ as above. Then, for a given γ, the string bases at each site j can be maximized independently, and both strings can also be maximized independently.

2. Since $\gamma < 0.5, \log(\gamma) < \log(1-\gamma)$ since $\log$ is monotonic and $\gamma < 1-\gamma$.

3. Let $s^*_{0,j}$ be such that the number of mismatched bases of reads X_{ij}, weighted by membership probability $T^{(t)}_{i,0}$, is minimized. I proceed by contradiction. Assume that $s^*_{0,j}$ does not maximize W (i.e. the string that does maximize W has some other base b at position j). Then changing $s^*_{0,j}$ to b must increase W. However, let:

   $$W_{j0}(s_{0,j}) = \log(1-\gamma)\Big(\sum_{i \in V_j \wedge X_{ij}=s_{0,j}} T^{(t)}_{i,0}\Big) + \log(\gamma)\Big(\sum_{i \in V_j \wedge X_{ij} \neq s_{0,j}} T^{(t)}_{i,0}\Big) = \log(1-\gamma)w_j + \log(\gamma)w'_j$$

   Since $s^*_{0,j}$ minimizes the weighted number of mismatches, changing from $s^*_{0,j}$ to b will move some terms $T^{(t)}_{i,0}$ from the left sum to the right, which has a smaller coefficient, that is, w_j must decrease by some ϵ, and w'_j must increase by the same amount:

   $$\begin{aligned} W_{j0}(s^*_{0,j}) - W_{j0}(b) &= \log(1-\gamma)w_j + \log(\gamma)w'_j - \log(1-\gamma)(w_j - \epsilon) - \log(\gamma)(w'_j + \epsilon) \\ &= \log(1-\gamma)\epsilon - \log(\gamma)\epsilon > 0 \end{aligned}$$

   which is a contradiction, since by assumption, $W_{j0}(s^*_{0,j}) - W_{j0}(b) < 0$.

Unsurprisingly, for all relevant values of γ, the strings that maximize the Q function are those that minimize the weighted number of mismatches to the reads. On the other hand, given two strings, the value of γ that maximizes Q is standard:

$$\gamma^{(t+1)} = \frac{\sum_{i=1}^{n} m_{i,0}T^{(t)}_{i,0} + m_{i,1}T^{(t)}_{i,1}}{\sum_{i=1}^{n} l_i T^{(t)}_{i,0} + l_i T^{(t)}_{i,1}}$$

where l_i is the length of read i, and $m_{i,k}$ is the number of mismatches of read i to string k.

3.3.5 EM regularization

I utilized regularization to restrict the estimated strings to be within a given range of distance to one another, as well as to be constrained to a reference panel. I reformulate this problem with the constraints. In this case, I assume s_0 is fixed:

$$s_1^{(t+1)} = \arg\max_{s_1} W(s_1)$$

Subject to:

$$h(s_1^{(t+1)}, s_0) > d$$

where h is the hamming distance. This is equivalent to a MAP-EM using a prior probability of zero on any strings that do not satisfy the constraints.

I first examine the distance-based constraint for $h(s_1^{(t+1)}, s_0) > \epsilon$. Let $\mathcal{S}$ be the set of all admissible strings. Let s^* be the string for which Q is maximized. Swapping any base of s^* that does not match s_0 so that it does match reduces Q. Let $\mathcal{J} = \{j : s_{0,j} \neq s_{1,j}\}$ be the indices where the strings differ, $\mathcal{W} = \{W_{j0}(s_{0j}) : i \in \mathcal{J}\}$ be the set of terms for each genome position when maximizing $W(s_0, s_1|\gamma, \theta^{(t)})$. Finally, let $\mathcal{I}_d$ be the indices with the d largest contributions to W_{j1}. Then the constrained maximum:

$$\arg\max_{s_1} W_{j1}(s_1|\gamma, \theta^{(t)}) = \begin{cases} s_{1j}^* & j \in \mathcal{I}_d \\ s_{0j} & otherwise \end{cases}$$

To see why this is maximal, take any other position j (or set of positions) where s_1 varies from s_0, but where $j \notin \mathcal{I}_d$. If any $j' \in \mathcal{I}_d$ is swapped for j, then $W_{j1}(s_0, s_1|\gamma, \theta^{(t)})$ is smaller, and so $Q(\theta|\theta^{(t)})$ is not maximized. I use the same procedure for choosing reference-constrained values for s_1.

3.3.6 Likelihood calculation

In order to compute the full log likelihood, per-read likelihoods must be calculated:

$$\log P(X_i|s, \pi, \gamma) = \log \Big(\pi P_0(X_i|\gamma) + (1-\pi) P_1(X_i|\gamma, s)\Big)$$

Both P_0 and P_1 are, assuming a position-homogeneous mutation rate, a function of the number of mismatches to the zeroth and first cluster sequence, respectively. Consider an update from s to s' and re-computation of the likelihood. If s' differs by a single base at some position, then only the reads that overlap that position will have a new likelihood term. As such, by maintaining an index V, such that V_{jc} gives a list of reads that have base c and position j, for a sequence of substitutions from s to s', we can query this index and update the likelihood components for each of these reads without iterating over all N reads.

Further, this loop can be computed in $O(V_{jc}J)$ instead of $O(V_{jc}L)$ time, where J is the number of changes in a transition between s and s', since, for each read only constant time is required for an update: consider a new string s' with one substitution at position j that is covered by read X_i. Let $p_0 = P_0(X_i|\gamma)$ and $p_1 = P_1(X_i|\gamma, s)$. If $X_{ij} = s_j$ (using a global index for j), then $p'_1 = p_1(\gamma/(1-\gamma))$. So, if there are J substitutions in a transition from s to s', only J updates, and evaluation of the mixture likelihood, are recomputed. If the parameter π changes, then neither P_0 nor P_1 need to be recomputed. For the slowest parameter update, γ, all components must be updated. To accelerate this case, instead of re-counting the hamming distance between a given read and s, I store the number of mismatches for each read in an array N, since I assume position-independent mutations in practice. After initialization, this array is dynamically updated on the transition from s to s'.

3.3.7 Masking sites

For the non-reference panel estimation, in order to decrease the cost associated with both sampling γ, as well as the number of sites L, a form of heuristic feature selection was implemented by site masking. Sites are passed through a preliminary filter which attempts to fix those which have low variation, that is those sites that are heuristically judged as unlikely to house SNVs. For example, if the set of sequences of a sub-clade has a diameter (with respect to the Hamming metric) of 50, I expect at most 50 sites of the alignment to require estimation. For those sites with the lowest variation, a point estimate of γ can be performed via specification of the `--point_estimate_gamma` argument. Secondly, only positions, and respective bases, which are likely to be SNVs, are estimated during estimation.

Three modes of site masking were implemented (`--basic_mask`,`--window_rank_mask`,`--rank_mask`). Site fixation was implemented per-window. For the basic mask, and including in the other filters, only sites that surpass a depth of 20 were considered. For the rank mask, the top 100 sites are unmasked, but not necessarily distributed into windows. For the window rank mask (influenza simulations), 100 sites were allowed to be unfixed across the HA sequence (approximately 1800 sites), distributed in 200 base windows, where the locally top sites are unmasked.

3.3.8 Decision rule

I utilize a modified likelihood ratio test, as in (Chen and Cheng; 2000) for $H_0 : \pi = 1, s_0 = s_1$, with the statistic:

$$-2(l_n(1, \tilde{\gamma}, \hat{s}_1) - l_n(\hat{\pi}, \hat{\gamma}, \hat{s}_1))$$

Which I assume to be asymptotically $\frac{1}{2}\mathcal{X}_0^2 + \frac{1}{2}\mathcal{X}_1^2$ distributed (Lindsay; 1995). Since the likelihood is not continuous w.r.t. the parameter s_1, which is discrete, I fix the estimated s_1.

3.4 Results

3.4.1 Simulations

Simulated populations

Figure 3.1 gives summary statistics for the simulated populations. The mean hamming distance between selected reference pairs was 60.23 (standard deviation 42.59). The sample mean of expected distance between quasispecies and their generating sequence s_0 for the major population was 0.085 (standard deviation 0.0388), which was approximately the same for the minor populations as expected. I note that this is the mean amongst samples, where each sample itself represented a probability mass function (and hence we refer to mean of expectations instead of mean of sample means). The mean sample mean coverage was 579.80 (standard deviation 26.29). The mean sample mean hamming distance of reads to the consensus was 5.09 (standard deviation 1.9). As such, the number of mismatches in reads was relatively high on average.

Estimation for simulated populations

For the unconstrained estimation, where $d(s_0, s_1) > 5$, but bases are not constrained to a reference panel, the mean absolute error for π, $|\pi - \hat{\pi}|$ was 0.026, and for γ, was 0.00027. As such, the point estimates for these parameters was reasonably good, despite the presence of quasispecies in these mixtures. For the LRT without quasispecies as shown in Figure 3.5 the chi-bar-squared distribution was a relatively good fit for λ. Figure 3.4 shows the distribution of $\log(\lambda + 1)$ for both simulations under the null $\pi = 1.0$ (orange), and mixtures (blue). This was still a relatively good fit for the chi-bar-squared. The LRT statistics for the two populations were well separated. As shown by the ROC curve in Figure 3.6, a 0.965 TPR could be achieved at a FPR at zero, since for many of the LRT statistics for the mixtures, the probability under the null was zero, so a decision based on any p-value greater

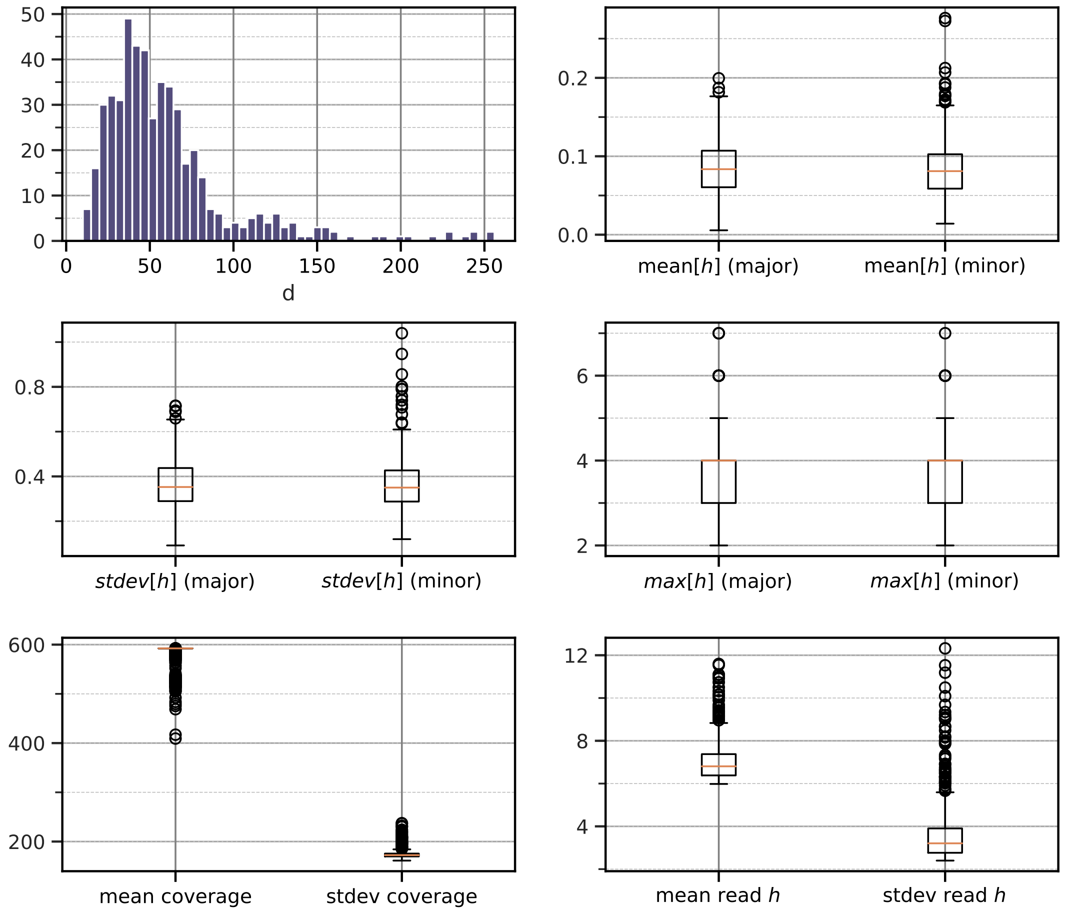


Figure 3.1: **Summary statistics for simulated populations.** From top left to bottom right: histogram showing distance of simulated major and minor sequence; the mean hamming distance of quasispecies to the cluster centroid; standard deviation of the same; maximum of the same; coverage mean and standard deviation; mean hamming distance between reads and the major sequence.

than zero would still result in a high TPR.

Figure 3.3 illustrates the error in $\hat{\pi}$ as a function of π for the panel constrained estimation. For the constrained EM algorithm, the mean error, 0.0178, was lower than that for the unconstrained estimate, and appeared to be less biased.

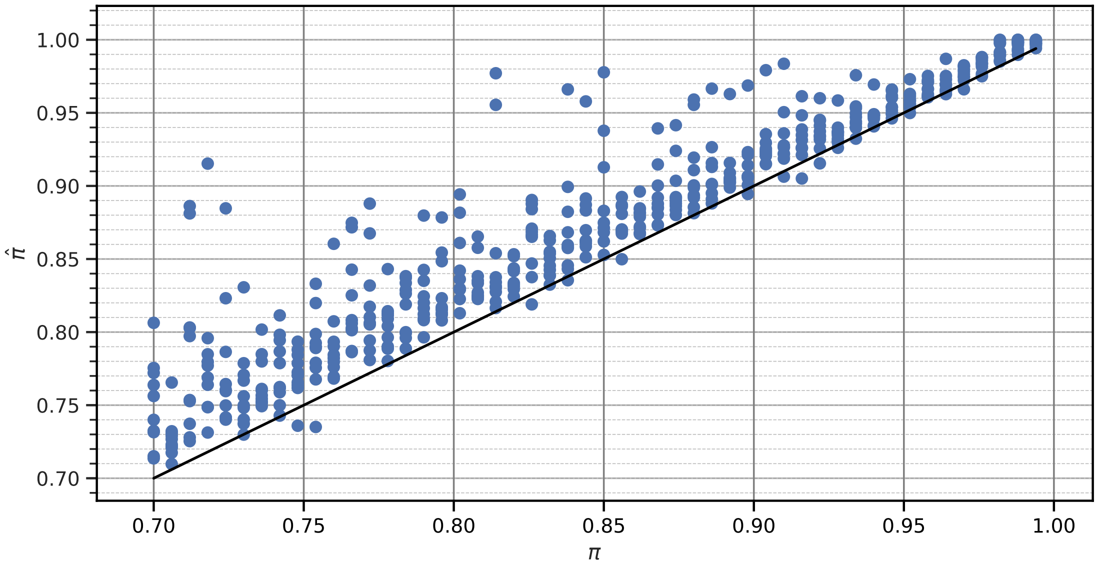


Figure 3.2: π **versus** $\hat{\pi}$ **for unconstrained estimation.** Mean absolute error $|\pi - \hat{\pi}|$: 0.026. Black line $x = y$.

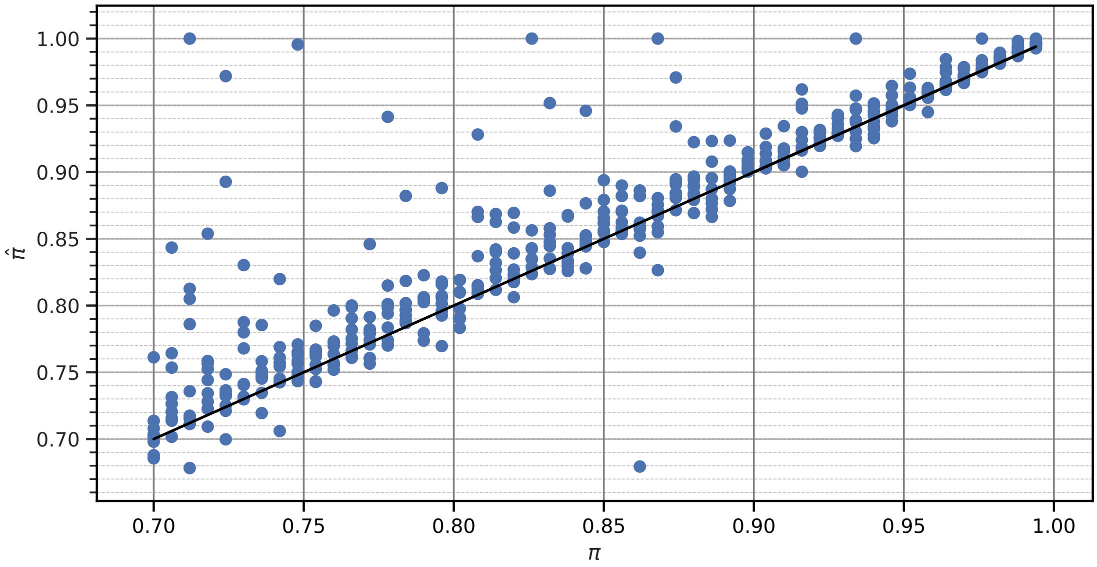


Figure 3.3: π **versus** $\hat{\pi}$ **for reference-constrained estimation.** Mean absolute error $|\pi - \hat{\pi}|$: 0.0178. Black line $x = y$.

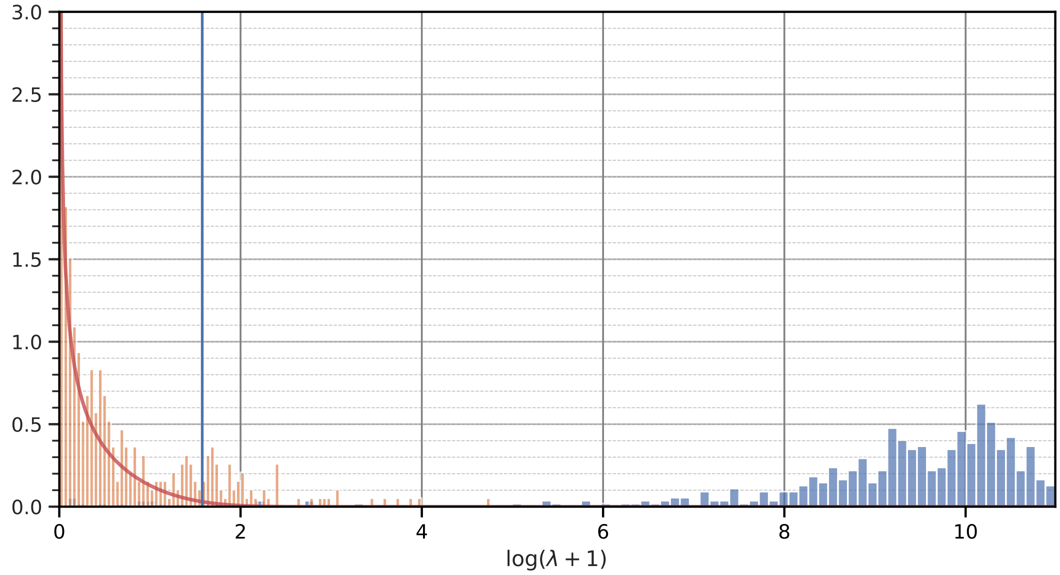


Figure 3.4: **Log of likelihood ratio statistic for simulation (with quasispecies) (blue) versus null simulation A3 (with quasispecies) (orange).** Orange line: theoretical chi-bar distribution. Vertical blue line: 95 % theoretical density. The two populations are well separated.

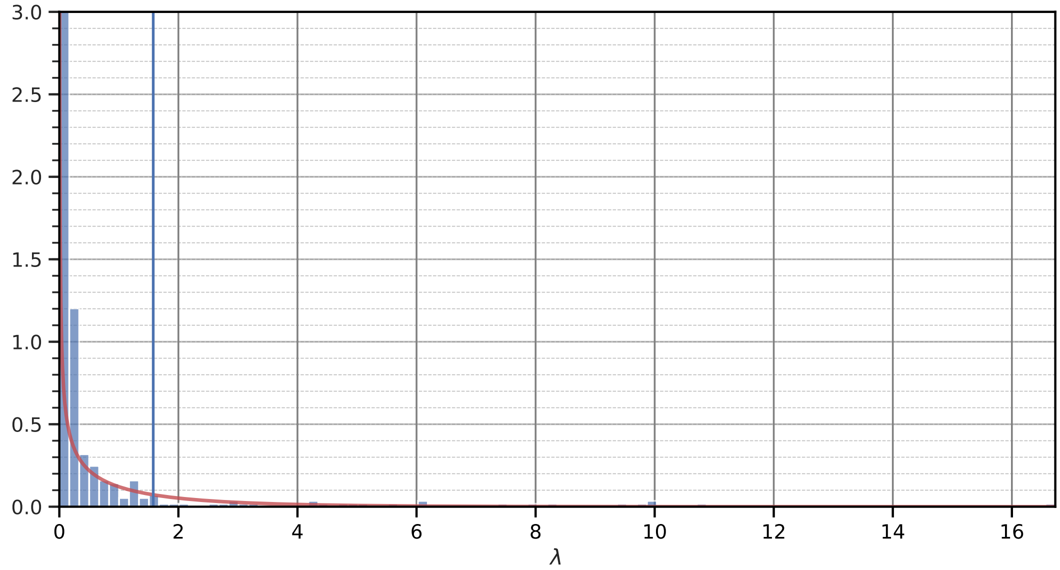


Figure 3.5: **Likelihood ratio statistic for null simulation without quasispecies.** Orange line: theoretical chi-bar distribution. Vertical blue line: 95 % theoretical density.

3.4.2 Real data

In order to deal with uneven coverage, for real datasets, a subsampling strategy across the genome was utilized, where reads were sequentially sampled randomly at each position from left to right, repeatedly. For real datasets, forward reads were mapped against a single reference HA sequence. Resultant alignments were mixed in pairs. Figure 3.7 gives the $\hat{\pi}$ against π for synthetically mixed

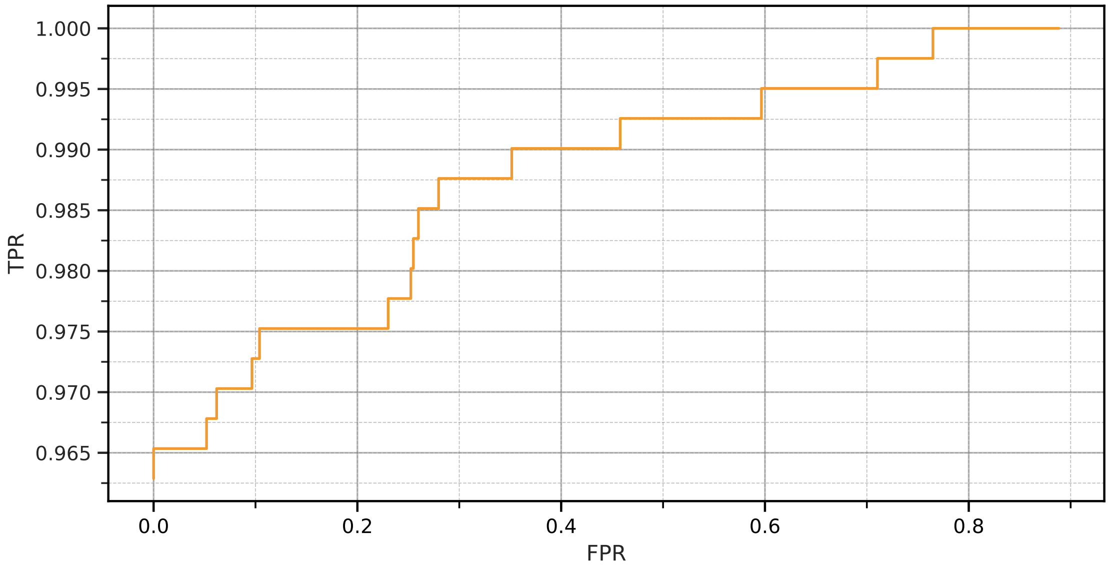


Figure 3.6: **ROC curve for detecting mixtures**. Here, p is varied. AUROC: 0.994, min: 0.965. Most of the observed mixtures have test statistics with probability near zero under the null, causing the intercept.

real influenza alignments. Estimation was performed with a panel of full length assembled HA sequences. The mean error was 0.0304, which was slightly worse than the simulated datasets. This was somewhat expected, since factors such as coverage vary more widely for real data.

In a test of 97 real read sets, 10 were found to have significant signatures of cross-contamination with an estimated sequence different to that of the consensus and $\hat{\pi} > 0.5\%$ (samples 101, 103, 111, 150, 166, 171, 225, 26, 36, 42) and $\lambda > 6$ (chosen *ad hoc* by consideration of Figure 3.9. The number of unmixed samples with extreme values of λ was greater than expected. This may be due to the fact that, in these datasets, technical or quasispecies mutations/errors can occur that are present in the sequence of some other sample, even if at very low proportion. Of these samples, 4 were estimated to have a proportion greater than 10%. In order to investigate further, variants were called with LoFreq (Wilm et al.; 2012). Of the 10 flagged samples, all except 3 had SNVs at every positions of difference between their consensus and the estimated sequence, indicating that they may have been contaminations. For two of the sequences, 42 (1/4) and 166 (9/23), only some of the differences were detected by LoFreq. This may be due to reduced coverage of a contamination, or due to coinfection with a novel sequence not present in one of the assembled samples. 2 samples detected as significant had a reference assembly that was different to their consensus (either due to a misassembly, or perhaps due to read sub-sampling); since the used version of `codetectem` relied on this assumption, they were excluded from this particular analysis. Future updates to `codetectem` could be used to account for this edge case. As shown in Figure 3.8, many samples had a large number of variants exceeding a 1% threshold, indicating that a method based solely on detection of variants alone may not be straightforward.

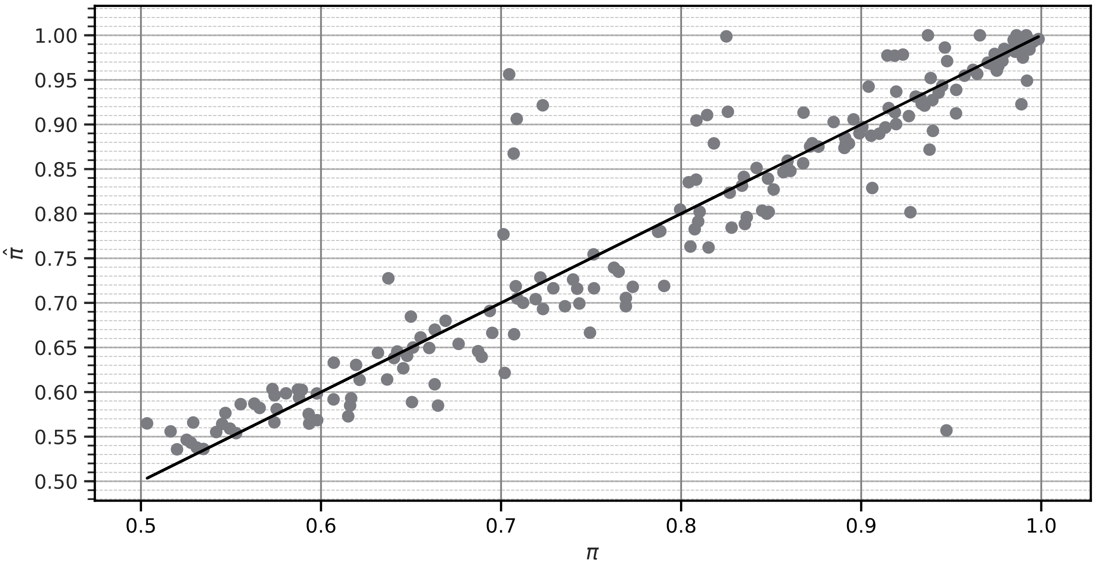


Figure 3.7: π **versus** $\hat{\pi}$ **for artificial mixtures of real data.** Mean error $|\pi - \hat{\pi}|$: 0.0304. Black line: $x = y$.

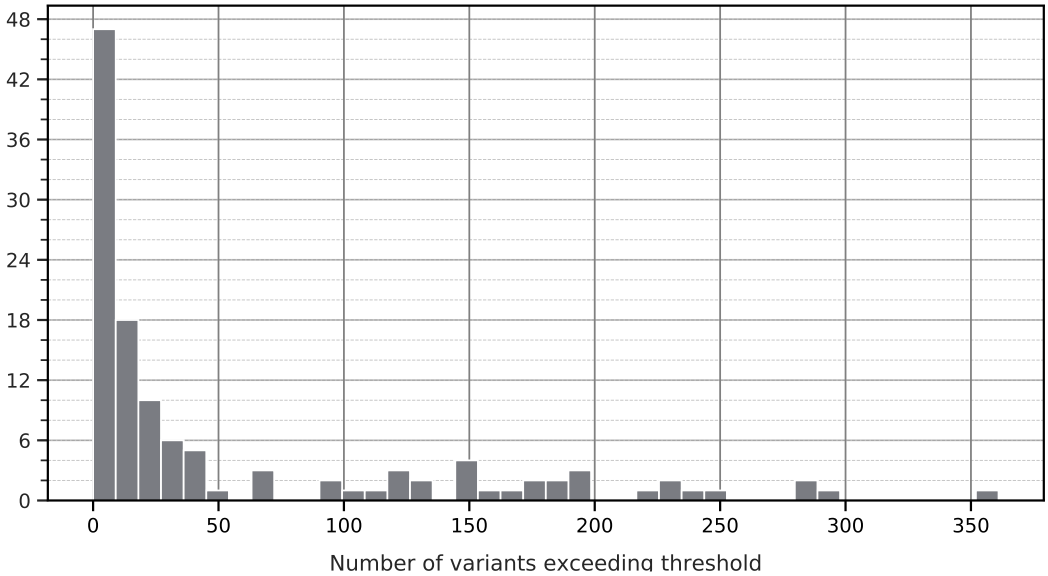


Figure 3.8: **Number of variants called by LoFreq with a proportion greater than 1%.** Mean: 57.53, 75% percentile: 95.

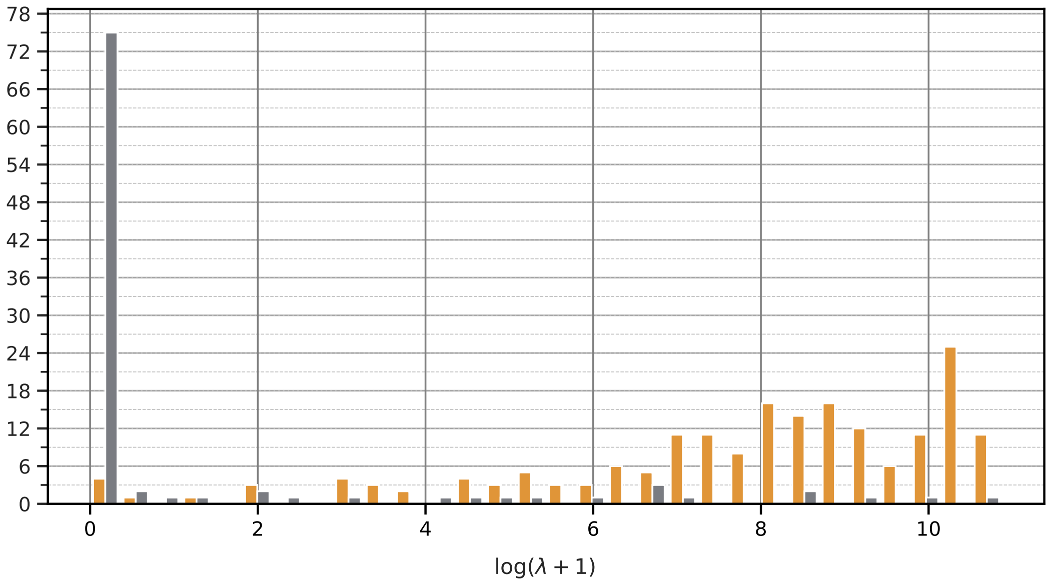


Figure 3.9: **Histogram (density) showing $-2(L(\theta_0) - L(\hat{\theta}))$ for real datasets (grey) and simulated mixtures of real data (orange).** Although it is not known how many datasets were contaminated or coinfected in reality, it is believed that the number of sequences with large values ($\lambda > 4$, are high $\lambda >> 4$ are extremely improbable under the null) is much higher than expected. Other reasons could exist for this, such as poor model fit (perhaps due to minor variants and extremely biased coverage). Some of the mixed samples had $\lambda = 0$; however, some of these mixtures had extreme differences in the number of reads (several orders of magnitude), and so subsampling 5000 reads could easily result in a dataset that is in fact, not a mixture.

3.5 Discussion

3.5.1 Overview

In principle, a method based on EM has several advantages over simply attempting to examine samples with a large number of variants called by tools such as LoFreq, such as inference of cluster membership probabilities, which allows for reads to be separated out or cleaned for separate assembly. In principle, by calculating the likelihood for full reads, I also retain information within reads that is lost by reducing them to an array of multinomial counts. For example, a 2-base array with A/G ratio 0.8 at position j, and a C/G ratio of 0.9 at position j' may appear to support haplotypes AC and GG in a proportion 0.85. However, with skewed coverage, the variants could in fact be AG and GC. Whole reads, however, if they cover j and j', can identify the correct haplotypes. Furthermore, my method explicitly models the proportions of the sequences, which is more difficult if only considering the marginal variants at each position, especially with thresholding. Finally, my model-based approach also gives some theoretical justification for classifying mixtures without true coinfections as training data, and reports the proportion π.

3.5.2 Simulated populations

I simulated populations with additional quasispecies variants, which were designed to be present at a combined frequency of up to a few percent, such as for influenza. Naturally, the simulation of these variants was not biologically realistic. The mean mismatch rate in simulated reads was relatively high, at 3.5%; real datasets had a mismatch rate slightly lower, but were up to 4%. As such, I believe that the simulated populations did have relatively large sequence variation. However, in real populations, variation is not randomly distributed, as in simulations; error during the RT-PCR process can lead to the appearance of minor variants. Furthermore, coverage for real sequences is not uniform, as it was in simulation, but is often highly skewed. Deviation from the null in many of the unmixed real-datasets could possibly be explained by these two factors.

3.5.3 Model

Similarly, the model I used did not explicitly account for quasispecies variants, although I hoped that the model would be flexible enough via the error rate γ. It is possible that a more sophisticated model could account for this better, especially in other viruses where quasispecies variants are present in higher quantities.

I chose to model the full read likelihood, including its order, instead of summing bases at each position in the alignment and employing a multiomial model as others have done previously. In principle, using multinomial counts does not account for the correlation between bases, and cannot as easily identify, for example, a sequence with two SNPs, versus two sequences with one SNP (and cannot make use of their co-occurrence in reads). Although this approach is compuationally demanding, I have developed an algorithm that reduces the complexity to be linear in the number of bases aligned to each position that will change; this could make the difference between 2×20, for a depth of 20 at both positions, to $2 \times 20 \times 200$ calculations. I found that using the naive approach was computationally infeasible (hours), whereas, for 5000 reads, my unoptimized Python implementation generally converged within a few minutes.

Another weakness of my method is that per-base error probabilities are not incorporated. Future work could incorporate base qualities, and additionally, prior information of variability across the length of the influenza genome.

3.5.4 Inference

For simulated populations, performance was good, even with the presence of simulated quasispecies, for both unconstrained and reference-constrained estimation. Without quasispecies, the chi-bar distribution for data simulated under the null was a reasonably good fit. With quasispecies, a second mode could be observed. However, in comparison to simulated mixtures, the test statistic λ was separated by several orders of magnitude between the two populations. A likelihood ratio test based using my simple model as such would likely be too sensitive, and as such, the parameter p would have to be tuned to specific use cases.

For mixtures of real data, estimates of π were good; the mean error was just 2%. For application directly on real data, results were harder to assess; since I cannot guarantee that these samples are not contaminated, the unmixed datasets could not be assumed to have been generated under the null. As such, I ran reference-constrained `codetectem` directly on these datasets, and found that many of them had large values of λ. This led us to impose the dual constraint $\hat{\pi} < 0.99$ and $\lambda > 50$. Furthermore, I used LoFreq to assess whether the positions that varied between the two sequences were also present as SNVs, which in most cases, they were. I recommend that LoFreq is also incorporated into the pipeline. For one sample in particular (sample 2), 223 SNVs were identified; although `codetectem` identified this sample, it is unlikely that the sequence estimated as s_1 was actually the contamination sequence. In this case, it is possible that sample 2 represents a coinfection, or a contamination between influenza A and influenza B (which was not assessed).

3.5.5 Use of fixed reference panels

The use of a fixed reference panel allows us to screen for contamination between specific candidates, with potentially higher performance than trying to model any sequences in the unconstrained problem. For example, for the real datasets, I used assembled sequences from these samples as a panel, allowing us to screen for contamination within a run.

3.5.6 Software considerations

The first version of `codetectem` was implemented in Python 3, which is slow. Future versions should be implemented in C++ or similar for a speedup of several orders of magnitude, which would also allow for a larger number of reads to be sampled. Although `codetectem` was implemented with software design principles in mind, and some test coverage, this should be increased in future releases; it is possible that some outliers in these results are the result of unexpected behaviour in implementation (bugs). Furthermore, my update algorithm optimizations could have benefit in other developed software. I note that this is only possible because I use a discrete string representation of the coinfection; for approaches using arrays $v \in [0,1]^{L\times 4}$ to represent populations, optimized updates would not be possible.

3.5.7 Pipeline implementation

These results show that `codetectem` could be useful, but I recommend that LoFreq also be used for examining the SNVs. Furthermore, I suggest that a quick, pairwise mixing experiment is performed per run (as in 3.9) to check performance for a given dataset, especially if different laboratory protocols or input viruses change.

3.5.8 Future work

Future work could focus on the following goals: the software should be implemented in C++ for speed, such as to increase the read sample size; implementation of read pairs, instead of using single, unpaired reads; results should be compared to what can be achieved with ShoRAH and other state of the art tools for estimating virus population structures in short reads; define a model with over-dispersion so as to accommodate quasispecies and RT-PCR errors that introduce structure into the data; test on other respiratory viruses such as SARS-CoV-2.

Chapter 4

Virus pangenome range query and nearest neighbor search for large-scale phylogenetic pipelines

4.1 Abstract

4.1.1 Background

For modern phylogenetic pipelines that process thousands of pathogen genomes, especially those used during the COVID-19 pandemic, range queries and nearest neighbor search is often necessary, such as for identification of possible linked cases. In order to draw trees, multiple sequence alignment must be performed, which can either be a full MSA, which can be slow and memory intensive, or a pseudo-MSA, where sequences are aligned to a reference, which is less accurate and may still be slow, depending on cost functions used for alignment. If the edit distance is used as the cost function, fast algorithms exist. From a given alignment, the SNP distance can be computed, which has a simpler correspondence to evolutionary distance than an alignment score. Furthermore, unknown bases (Ns) present in appreciable quantities can complicate calculation of the SNP distance. Here, I explore the task of fast nearest-neighbor search and range query for virus genomes.

4.1.2 Methods

Firstly, a fast method for computing SNP distances for SARS-COV-2 genome sequences, called `NBRFIND`, was developed. I assessed SNP distances calculated from alignments computed under the edit distance and a scheme with higher indel costs. I then extended the fast diagonal algorithm of

Ukkonen to accommodate: higher indel costs; simultaneous calculation of the SNP distance, to prevent explicitly computing the alignment; a probabilistic mismatch score that can incorporate unknown bases. Secondly, I performed preliminary work to explore search strategies. I briefly explored i) the utility of radix trees to eliminate redundant calculations when computing the SNP score of a query against thousands of references and ii) kmer-based filtering.

4.1.3 Results

For computing SNP distances for SARS-CoV-19 genomes, `NBRFIND` was found to be nearly as quick on average as `edlib`, the state of the art for edit distance calculation, and more accurate, with respect to SNP distances calculated from the alignments. However, `edlib` offered much faster run-times for outlier sequences. Although our algorithm would still be too slow to permit a linear scan of many queries against hundreds of thousands of references, the task can be easily parallelized. Preliminary results for both k-mer filtering and a radix-tree based algorithm did not indicate sufficient benefit for further work. More extensive future research could examine the best way to perform range queries by filtering or pruning. Currently, I suggest the best approach to finding epidemiologically relevant sequences still relies on a MSA or pseudo-MSA for pre-alignment, which can then be reused. Lastly, given the heuristics employed to improve runtimes, I conclude that practically speaking, an approach based on seeds (such as MUMs) may be superior.

4.2 Background

During the COVID-19 pandemic, genomics became a core part of the UK's response (The COVID-19 Genomics UK (COG-UK) consortium; 2020), in particular to aid in several aspects of surveillance, including but not limited to: studying transmission chains; epidemiological dynamics and importation, including spatial dynamics; identification of mutations that may alter viral fitness; integration with other epidemiological data for understanding impacts on health and treatment. At the forefront of these efforts was the *$*20 million COVID-19 Genomics (COG) Consortium, whose efforts directly fed into the UK Scientific Advisory Group for Emergencies (SAGE). Within COG, data is incorporated into a phylogenetics pipeline which aggregates thousands of SARS-CoV-19 genomes into a large alignment phylogenetic tree (Nicholls et al.; 2020). Whilst these trees are highly curated, the pipeline involves strict filtering for quality, including by genome coverage, followed by multiple sequence alignment (MSA) and tree building. These processes can be slow for large sequence collections, and relying on weekly upload schedules for data. As a result, for participating centres, waiting for such a pipeline to complete before performing data analysis (several days) may not be sufficient for epidemiological investigations. Furthermore, for other future applications, such predictive applications of phylodynamics, where inferences must be made in near real-time, an additional delay of several days would be unacceptable. In principle, many genomic analyses do not require such a large collection of genomes. In cases where, for example, only a small subset of samples is required, such as cluster investigation, or phylodynamic studies into smaller sub-regions, only an input sample set, and their nearest neighbors, are required. However, without a MSA or a tree of all samples, the latter is difficult to ascertain.

Two related problems, the *range query* (Hu and Lee; 2005) and *nearest neighbor search* (Dhanabal and Chandramathi; 2011) are applied to similar problems in a diverse range of fields, where query data points are compared to a collection of data points, of which only certain elements are of interest. Naturally, depending on the domain, methods vary widely. In the naive approach, for a query of length N and a reference set of length M, a linear scan can be performed. However, for large collections of data points, and more importantly costly distance calculation, this can be unacceptably slow. In low-dimensional spaces, performant, exact algorithms exist. In high dimensional spaces, exact methods may also perform poorly. As a result, many applications also resort to *approximate* methods, or *filtering*, where a fast filter can exclude unlikely candidate matches. In the field of string matching, extensive work has been performed addressing both range queries and nearest neighbor search under metrics such as the edit distance. While some algorithms are theoretically superior, in practice some algorithms may be more performant on different datasets, depending on pattern length, the edit distance itself, the text length, and the alphabet size (Navarro; 2001). Furthermore, many algorithms were developed with approximate string matching as an intended application, which is a form of local edit distance, instead of global distance. Furthermore, a lack of comparison for practical applications was noted by (Hyyrö; 2003).

For comparison of SARS-CoV-2 genomes for epidemiological purposes, a few unique problems are posed. Firstly, our algorithm must be of high sensitivity; for example, in outbreak investigation, all relevant samples must be found in order to discover any transmissions across country, or to find where the lineage was imported from. Secondly, since the genomes themselves are long strings, comparison can be slow, and many existing string matching methods (for example designed for words in the English language) are unsuitable. Furthermore, it is not clear what distance metric or dissimilarity function would be most suitable for comparing these genomes; the edit distance could be insufficient, due to the presence of a potentially low coverage, as well as inadequacies as a model for sequence alignment. However, it is closely related to other global alignments. In this case, the SNP distance provides a better proxy for evolutionary proximity; most phylogenetic tree building rely on variant sites, not indels.

General scoring schemes can penalize gaps over mismatches, in order to produce an alignment which aims to mimic realistic evolutionary processes. However, algorithms with arbitrary scoring schemes are slower in practice than the edit distance, for which faster algorithms exist. However, if we wish to calculate the SNP distance from an alignment (the hamming distance over an alignment ignoring gaps), the alignment must be accurate. For example, for the edit distance, whether an indel or a mismatch is used in the alignment is an arbitrary choice provided both have the same score. As such, I compute alignments for SARS-CoV-2 genomes with the edit distance scoring scheme, as well as one where where mismatches cost 1 and indels cost 2, which I refer to as the `2-indel` cost function. I then compare the resultant SNP distances of these alignments with one produced by a multiple sequence alignment with mafft (Katoh et al.; 2005), the gold standard in multiple sequence alignment used for virus phylogenetics.

4.2.1 Experimental objectives

Here, I assess several methods for exact calculation of the SNP distance between SARS-COV-2 genomes, which could then be utilized in clustering algorithms (such as in (Berman and Shapiro; 1998)). Specifically, I describe the following experimental objectives:

1. Formulate a good distance metric or dissimilarity function for use as a criterion for nearest neighbor search for SARS-CoV-2 genomes, with particular focus on the SNP distance. In doing so, compare SNP distances computed from global alignments based on the edit distance, and those based on a more realistic (but still fast) scoring scheme.

2. Develop a model used for fuzzy comparison of strings with many unknown bases (Ns).

3. Develop an algorithm for fast calculation of the SNP distance between SARS-CoV-2 genomes, and compare run-times with the state of the art for edit distance.

4. Assess the potential speed up in multiple comparisons offered by compression via radix trees, which are compatible with progressive global alignment, by examining the compression ratio of SARS-COV-2 genomes in a radix tree.

5. Assess the viability of basic filtering with kmer-based approaches.

4.3 Methodology

4.3.1 Estimating SNP distance from alignments

I compared the SNP distance calculated from alignment with the edit distance cost function (1 for mismatches, 1 for indels) as well as one based on the a cost function that penalizes indels twice much (1 for mismatches, 2 for indels), which I refer to as `2-indel`, to SNP distances calculated from a multiple sequence alignment computed with MAFFT, a state of the art multiple sequence alignment program (Katoh et al.; 2005).

Fuzzy SNP distances for sequences with Ns

Here I describe a basic probabilistic model to evaluate $P(d(x, y)) < \epsilon$. Let mutations be generated across the genome uniformly with probability μ. Let the number of mismatches with Ns be N, and the number of matching and mismatching alignment positions without an N be M (proper matches or mismatches). Let $d(x, y) = d + d_N(x, y)$ where $d_N(x, y)$ is the random variable denoting the number of mismatches in the N unknown alignment pairs, and d the observed mismatches in the M observed bases. Then, $P(d_N(x, y) < k - d|d)$, which is 0 if $d \geq k$, is given by:

$$
\begin{aligned}
P(d(x, y) < k|d) =& P(d_N(x, y) < k - d|d, M, N) \\
=& \sum_{z=0}^{k-d-1} P(d_N(x, y) = z|d, M, N) \\
=& \sum_{z=0}^{k-d-1} \int_0^1 P(d_N(x, y) = z|d, M, N, \mu)P(\mu|d, M, N)d\mu \\
=& \sum_{z=0}^{k-d-1} \int_0^1 P(d_N(x, y) = z|\mu, N)P(\mu|d, M)d\mu
\end{aligned}
$$

where z are the additional number of mutations that could have occurred in the N ambiguous matches. Also, the final equality results because $d_N(x, y)$ is conditionally (on μ) independent of M, and the prior is assumed to be dependent only on M and d. Since I use a binomial model for the distribution of mutations in unknown bases, I use a Beta conjugate prior, so:

$$
\mu|d, M \sim \mathrm{Beta}(\alpha + d, \beta + M - d)
$$

Which results in a Beta-binomial, so:

$$P(d(x,y) < k|d, M, N) = \sum_{z=0}^{k-d-1} \binom{N}{z} \frac{B(z+\alpha+d, N-z+\beta+M-d)}{B(\alpha+d, \beta+M-d)}$$

Diagonal alignment algorithm for `2-indel`**:** `uk2`

As a basis of approach I employ the diagonal algorithm of Ukkonen and Landau (Ukkonen; 1985a), as described previously, which is fast for similar strings. I adapt this scheme in order to incorporate the expression 'any base' (N), and to utilize the global alignment distance with indels twice as costly as mismatches. In order to achieve this, I utilize a variant on this recurrence:

$$\forall h > 0, L^h(d) = SLIDE_d\Big(\max \begin{cases} L^{h-2}(d+1)+1 & d < h \\ L^{h-1}(d)+1 & always \\ L^{h-2}(d-1) & d > -h \end{cases}\Big)$$

Furthermore, I permit 'N' to match any base in the `SLIDE` function. Practically, I also adjust the diagonal growing diagonal band, such that $-\lfloor h/2 \rfloor < d < \lfloor h/2 \rfloor$, since, for edit distance h, there can be at most $\lfloor h/2 \rfloor$ indels. In order to record the number of mismatches during computation, I additionally store an array M which records the number of mismatches, analogously to L, although it does not determine the optimal step, but serves as a record (in this case, multiple solutions with different numbers of mismatches is possible).

I used several heuristics in addition to improve run-times. When a diagonal SLIDE occurs that includes a sufficient number of matches, ending at (i, j), the alignment process is stopped and restarted at $(i+1, j+1)$. This method works on the basis that, if a SLIDE operation matches more than θ bases, then that diagonal is likely to be on the optimal path later on. For example, if a run of 1000 bases match, it is unlikely that they are not matches in the final alignment, given that these genomes do not have large repetitive regions. In this case, if SLIDE proceeds by more than θ bases, one can evaluate the number of mismatches on that diagonal to be reflective of the true SNP distance, and bail early if it exceeds a threshold k. For range queries, this can be used for early bailing during calculation. This procedure is similar to seeding with exact matches.

4.3.2 Preliminary exploration of nearest-neighbor search strategies

Radix Tree-based `2-indel` **algorithm:** `tree_uk2`

Several data structures can be used to index strings in order to accelerate batch computations. For the array L, clearly for any query P, any two sequences T_1, T_2 with common prefixes will share many of the same computations. As such, these prefixes can be reused. Let $T_1 = R + S_1$ and $T_2 = R + S_2$. In order to assess how much redundant calculation is performed with our chosen genome set, I built a Radix tree from a set of 5000 genomes, and recorded the total edge length as a function of the number of genomes.

k-mer filtering

Since the edit distance computation is expensive, and for the largest genome collections there may be up to 100,000 genomes of length 30 kilobases, a linear scan becomes too slow. As such, I assessed k-mer based dissimilarity filtering methods. 5000 random SARS-CoV-2 genome pairs were subjected to basic preprocessing previously, and the SNP distance was plotted against k-mer distances ($k \in (21, 50, 300)$). Additionally, I evaluated MASH with default settings for the same purposes.

4.3.3 Data used

I made use of SARS-CoV-2 genomes generated by the COG consortium (The COVID-19 Genomics UK (COG-UK) consortium; 2020). Furthermore, since the algorithm is, in practice, much faster for nearly-aligned sequences, I quickly perform the following pre-filter: i) genomes with more than 10% Ns were discarded; ii) the coding region were extracted; iii) any other ambiguous bases were converted to ‘N’s.

4.4 Results

4.4.1 Global alignment algorithm

Comparison of global alignment with uk2 and the edit distance

For 10,000 genome pairs, I compared the SNP distance computed from 1) a MAFFT MSA with default settings, 2) alignments using the double indel, single mismatch cost function, and 3) alignments using the edit distance. There was no difference in the SNP distances computed from the first two processes. For the edit distance alignment, the mean error (difference to the score calculated from

MAFFT MSA) was 0.357, which was driven primarily by outliers (see Figure 4.1). Whilst the edit distance had a low error, my approach had zero in the observed samples. As such, the chosen scoring function was superior to that obtained under the edit distance cost for computing SNP distances and, for this data, was found to be equivalent to a more realistic cost function, as used by default in MAFFT.

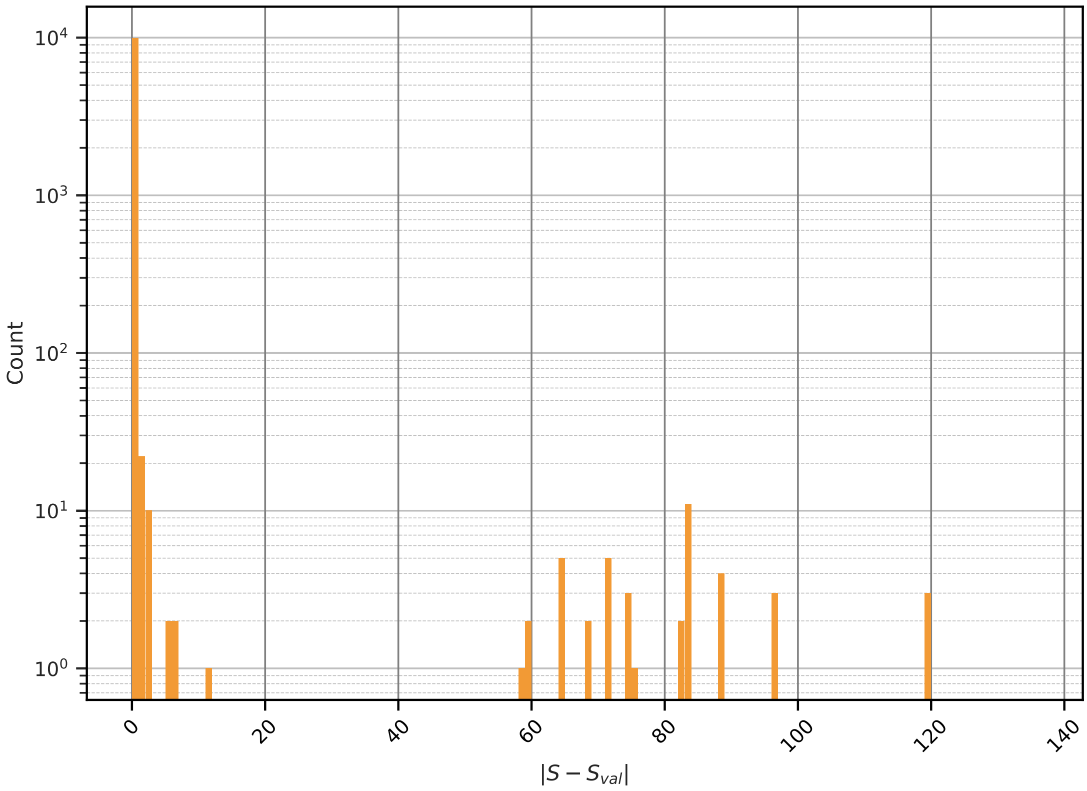


Figure 4.1: **Comparison of SNP distance S calculated from alignments minimizing the edit distance, in comparison to S_{val}, computed from a MSA with MAFFT**. Here, $10,000$ randomly sampled genomes were used, and the log score $\log_{10}|S - S_{val}|$ calculated. Whilst the edit distance produced alignments most often similar to MAFFT, occasionally outliers with present. The mean error in SNP distance was 0.357.

A basic run-time comparison was performed between the core of `NBRFIND` and `edlib`, as shown in Figure 4.2, where `NBRFIND` was allowed to terminate alignment early based on a maximum SNP distance of 5, such as in a range query. Since edlib, or any other basic global alignment algorithm, cannot perform early bailing based on the SNP distance, but only on the alignment score, the runtime for calculating the SNP distance from an edlib alignment will be at least as costly as the core alignment as it excludes post-processing. 5000 trimmed SARS-CoV-2 whole genomes with up to 10% gaps were queried with a randomly query, 1000 times. For a maximum distance of 5, Calculation from the edit distance was marginally faster, with a mean of 43.19 seconds, compared to that of 48.86

seconds for NBRFIND. This was expected, since NBRFIND has a more complex cost function, and also computes the SNP distance simultaneously. However, a few outliers that took around 500 seconds were observed, although this proportion was small; since the diagonal algorithm has complexity $O(nd)$, outlier sequences with errors or a large number of indels may take longer to align. In this case, edlib offered superior performance, although alignment post-processing was not included in the timings.

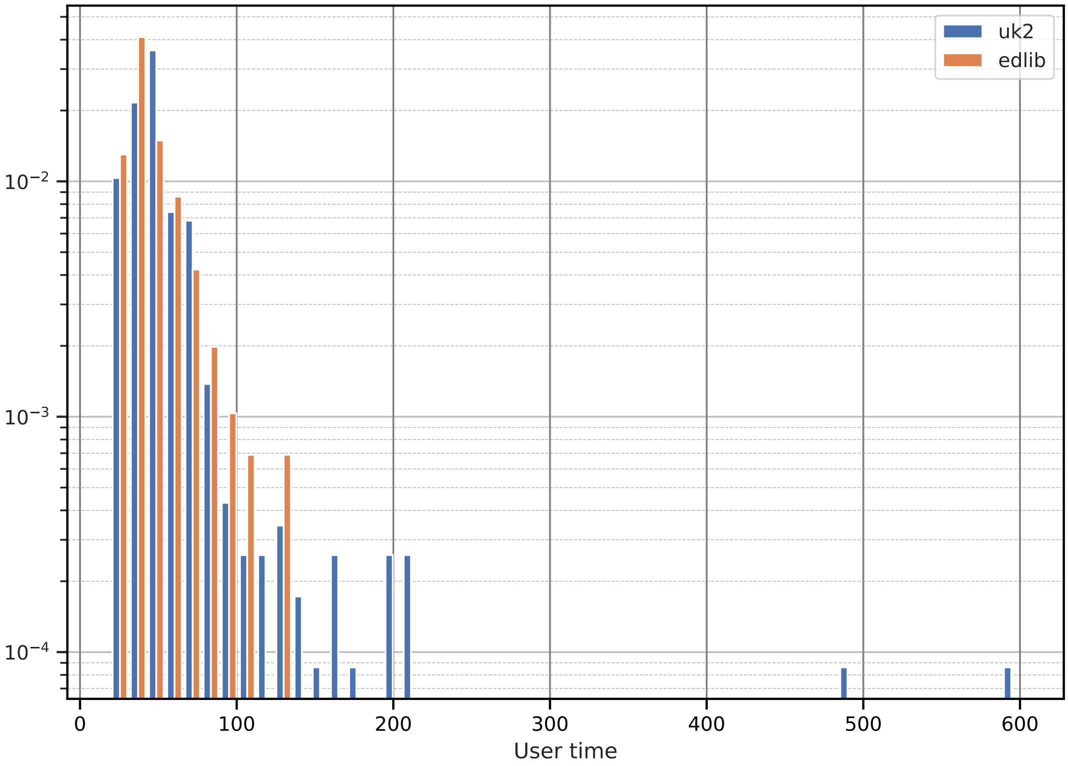


Figure 4.2: **Basic runtime comparison of the core algorithm of NBRFIND with that of edlib for 5000 comparisons.**

4.4.2 Preliminary investigation of nearest-neighbor search strategies

Filtering

I found that neither basic filtering based on the Jaccard index J, of k-mers ($k \in (21, 50, 300)$), nor the MASH distance, would be likely to offer the required sensitivity and specificity for practical application. Figure 4.3 shows the relationship between SNP distance and MASH distance for 1000 pairs of SARS-CoV-2 genomes. Since the objective was high sensitivity and specificity, I believe these results indicate both could not be achieved using the MASH distance.

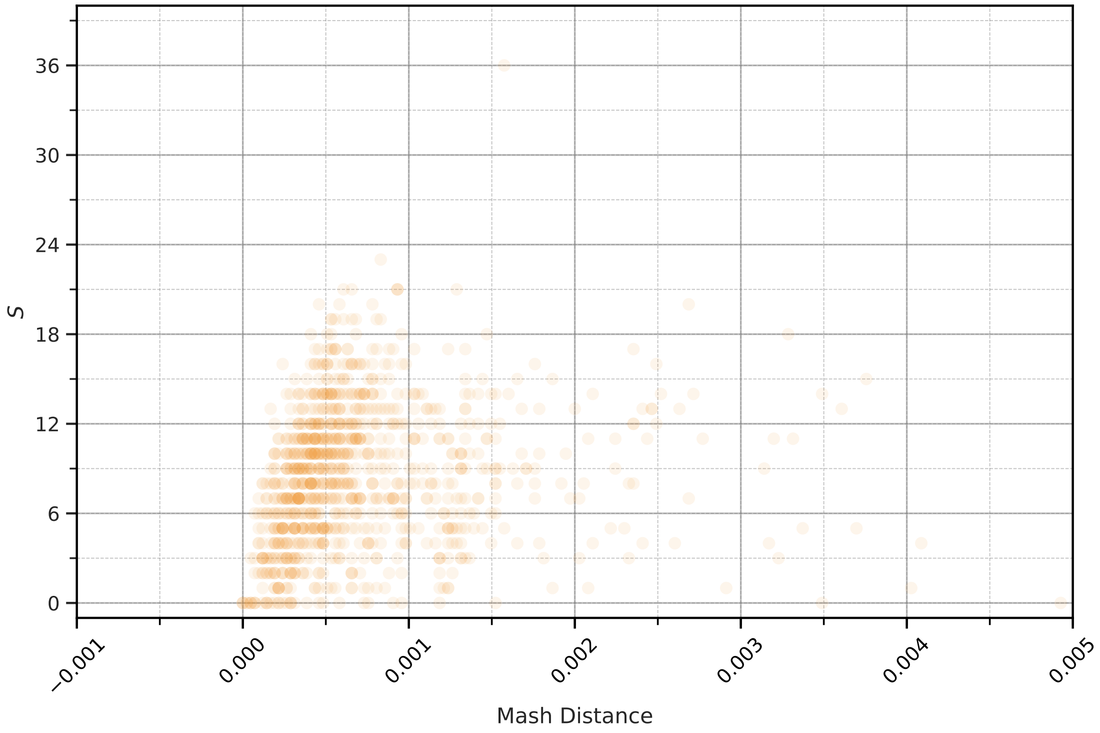


Figure 4.3: **MASH distance (default parameters) vs SNP distance.**

Data structures for sequence compression

Figure 4.4. gives the ratio of radix tree edge length to total sequence length as a function of the number of added genomes for 5000 genomes. As demonstrated, as N grows large this proportion approached just under 0.5 (in particular 0.489 at sequence 5000). As such, the compression ratio will likely not sufficiently improve run-times compared to the naive search, which is more easily parallelized.

4.5 Discussion

4.5.1 Alignment algorithm

The alignment score and algorithm developed in this chapter allowed for the calculation of SNP distances without explicitly producing an alignment identical to those produced from a state of the art MSA. Furthermore, I developed an approach to estimating the probability of a given SNP distance when unknown bases are included. In comparison to the state of the art in terms of alignment speed, `edlib`, our alignment performed well when allowing early exiting based on the SNP distance, but

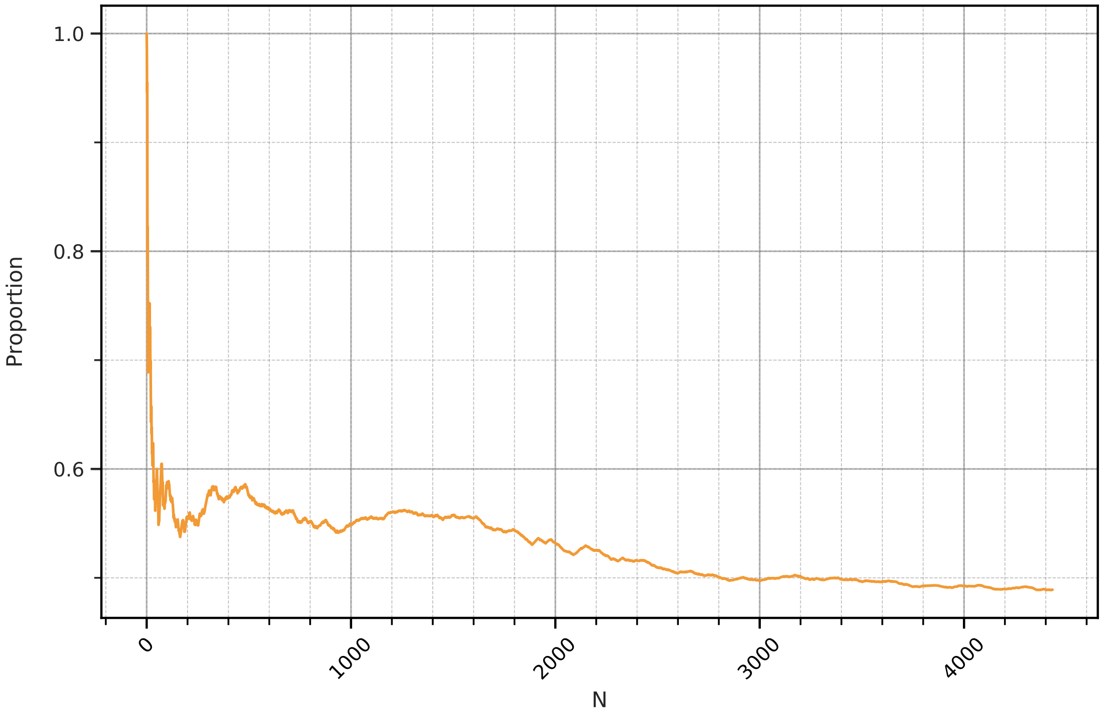


Figure 4.4: **Total sequence length versus Radix tree edge length for 5000 SARS-CoV-2 genomes.** Here, N denotes the number of genomes added to the Radix tree so far. As expected, the proportion rapidly decreases as sequences are added, since common prefixes are not repepated.

poorly in comparison for some outliers; on average, 5000 comparisons could be made in under a minute. Conventional approaches for alignment, including `edlib`, would not be able to perform early exiting based on the SNP distance, since this must be calculated from an alignment. However, whilst this performance may seem reasonable, only with parallelization could it be practically useful. For 60 queries, 5000 comparisons would likely take nearly an hour. Furthermore, a database of candidate sequences could be considerably larger than this, at least for SARS-CoV-2. Currently, the fastest method for nearest neighbor search would be to use a pseudo-MSA constructed by aligning to a single reference, and calculation of the hamming distance. For extremely large database sizes, however, even this calculation can be costly.

The developed approach may still be useful in tasks that require finding nearest neighbors, such as search strategies that involve pruning or partitioning the database, such as greedy clustering. Furthermore, it should be noted that, although my approach was implemented in C++, code optimizations could yield performance gains. Finally, subsequent to the project study, a diagonal algorithm for affine gap penalties was developed (Marco-Sola et al.; 2020). This development is an improvement on the algorithm developed in this chapter.

4.5.2 Search strategies

Initially, `NBRFIND` was intended to be used within an appropriate neighbor searching strategy. In metric spaces of low dimension, conventional methods can be applied. However, for high-dimensional spaces, many exact methods break down, and approximate methods are preferred (Ponomarenko et al.; 2014). Most k-ANN search algorithms make use of the triangle inequality to narrow down the search space. However, when this is violated, the choices are slim (Ponomarenko et al.; 2014). For any amount of Ns, assume $d(s1, s2) = d(s1, s3) = 0$. It is possible that $d(s2, s3)$ is still large. For some (dis)similarity functions, special solutions exist (Zhang and Srihari; 2002). I explored two techniques to improve on brute force: use of radix trees for reducing redundant computation of shared prefixes, and k-mer filtering.

I explored the radix tree for collapsing down common sequence prefixes. However, the compression ratio achieved was only about 50%. This was somewhat expected; if one assumes a new sequence is one mutation away from an existing sequence in the tree, then, assuming uniform distribution of mutations across the genome length, the number of new bases would be the suffix beginning at the mutation position, which would have expectation 0.5. On a test of 500 genomes, trie building took 5.022 seconds on average with a Python implementation. Although this building process is linear in the number of genomes, it is expected to require over 8 minutes to build the full genome set, although it could be considerably faster in C++. An additional problem for the radix tree is Ns; sequences with a large number of randomly distributed Ns are less likely to have long common prefixes.

Our preliminary investigation indicates that basic k-mer filtering with MASH would not offer good performance for filtering candidate sequences. Results for plain Jaccard distance are ommitted for brevity, but were found to be similar. The relationship between J and h, the alignment score was also not strong. Several causes could be responsible for this, including the fact that sequences are highly similar, and the presence of Ns (that is, partial genome coverage) in these sequences confounds similarity calculation. However, simply ommitting k-mers with Ns did not sufficiently change results, as expected. Statistical evaluation and ROC curves were not produced after graphical inspection of these scatter-plots as is was likely that no filtering threshold with these distances would be acceptable.

Conclusions

NBRFIND offered superior precision and similar speed to edlib in aligning SARS-CoV-2 genomes, although speed was poor for some outliers. Furthermore, for the naive linear scan nearest neighbor search, performance was not sufficient for single-threaded application. For access to a large number of threads, exact search could extract nearest neighbors in reasonable time. Furthermore, methods based on pseudo-MSAs are likely to offer better performance. Future study could be directed at approximate nearest neighbor algorithms to avoid the costly naive linear scan, although this goal may conflict with requirements for high precision.

Part II

Phylogenetic methods for respiratory virus surveillance

Chapter 5

An assessment of phylogenetic resolution for routine molecular epidemiology

5.1 Abstract

5.1.1 Background

Influenza A and other respiratory viruses are a severe global public health burden. For monitoring, sequencing has become key, augmenting traditional epidemiology. Traditionally, this comprised sequencing of HA and NA genes, but WGS has been increasingly adopted. Within the application of phylogenetic methods to virus epidemiology, phylogeny reconstruction and dating forms the foundation of increasingly complex phylodynamic methods. However, performances of these foundational processes are rarely quantified, especially with respect to the required number of genome segments. Molecular clock methods can be especially difficult to apply, since the rate of molecular evolution can be complex and variable, and computationally, inference is not straightforward. Routine application of activities such as estimation of tMRCA can perform poorly if not configured precisely. Furthermore, these methods can be very difficult or impossible to experimentally test after an inference has been made. As such, the adoption of molecular dating methods has not been as quick within routine epidemiology as other activities, such as cluster investigation.

5.1.2 Methods

Here, a set of analyses was performed to benchmark phylogenetic resolution and dating, and compare performance under different numbers of segments. Firstly, I compared bootstrap support for real data between trees constructed with 1 to 8 genome segments, as well as molecular clock estimates for HA-NA versus the whole genome, and evaluated inferences that can be made with the different datasets. I also compared three commonly used tools for molecular dating, treedater, treetime, and BEAST, on data from simulated influenza A epidemics, and examined the accuracy for small time periods that can be found in molecular epidemiology.

5.1.3 Results

Whole genome sequencing was found to provide increased resolution for phylogenetic reconstruction in terms of bootstrap support, allowing identification of outbreak samples with high support from a single season. It was found that, for sequencing, at least 5 of the 8 segments provide good phylogenetic resolution. With only HA and NA, resolution was insufficient for many downstream phylogenetic tasks on the scale of a single season. I found that molecular clock estimates were also improved, with less variance, and estimates were more consistent with previous studies. In simulation, I found that on the scale of a single season (a few months), molecular clock estimates were reasonable but not precise; mean error in estimated TMRCA was on the order of a third of the epidemic time-scale.

5.2 Background

Although until recently, most sequencing efforts for influenza have been directed to HA alone, whole genome sequencing has begun to emerge as a powerful tool in the study of influenza virus populations, since the whole genome contains more information. Laboratory protocols have become increasingly standardized and optimized (Zou et al.; 2016)(Wüthrich et al.; 2019)(Lee et al.; 2016) to enable reproducible single-reaction RT-PCR and next-generation sequencing (NGS) (Zhou et al.; 2009)(Zhou et al.; 2014) by public health laboratories. Single-molecule sequencing platforms such as the MinION (Imai et al.; 2018b) have also been explored. As a result, whole-genome sequence data is beginning to accumulate alongside the large number of HA and NA gene segments in public databases (Bao et al.; 2008a)(Shu and McCauley; 2017), which has been used to provide useful insights into the evolution and spread of influenza, as well as increasing rapidity and accuracy of analysis, with examples including: defining outbreaks (Meinel et al.; 2018b)(Houghton et al.; 2017)(Houlihan et al.; 2018a); study of intra-host quasispecies populations (Meinel et al.; 2018b);

examination of amino acid substitutions associated with antigenicity, antiviral susceptibility (McGinnis et al.; 2016a), severity (Galiano et al.; 2012) or hospitalization (Mishel et al.; 2015); identification of reassortment events (Goldstein et al.; 2018)(Holmes et al.; 2005b)(Oong et al.; 2017); phylogenetic clustering of severity (Goldstein et al.; 2018); characterization of co-circulating lineages (Baillie et al.; 2011)(Holmes et al.; 2005b); analysis of spatio-temporal dynamics (Lewis et al.; 2015)(Baillie et al.; 2011).

However, whilst laboratory methods have become widely applicable, best practices for the analysis of the data generated have not been standardized, and importance has not been demonstrated for all applications. Crucially, benchmarking of the improved resolution provided by whole genome information, and when it is useful or necessary, have not been performed. A wide range of approaches have been used for reconstruction of trees, cluster definition, and hypothesis testing, that do not always make optimal use of the data available. Tree reconstruction has often been performed with basic methods (Tamura et al.; 2004)(McGinnis et al.; 2016a)(Meinel et al.; 2018b). Methods for inferring quasispecies variants have been simple (Meinel et al.; 2018b), and some concerns have been raised with regards to hypothesis testing of the association of molecular features with epidemiological quantities, particularly with regards to sampling procedures (Goldstein et al.; 2018) and the trade-off with statistical power(Lewis et al.; 2015). Recently, methods for phylogenetic inference and modelling have become increasingly sophisticated, including methods for controlling for sample non-independence in testing significance of tip associations (Felsenstein; 1985)(Ives and Zhu; 2006)(Parker et al.; 2008), methods for phylodynamic analysis with large datasets (Sagulenko et al.; 2018), inference of reassortment events (Nagarajan and Kingsford; 2010), and complex phylogenetic modelling platforms such as BEAST (Bouckaert et al.; 2014). To accompany this increasing complexity, it is important that foundational tasks such as tree building and clock-rate estimation are benchmarked.

Some previous research has attempted to demonstrate the utility of WGS over single-gene approaches. Most often these include the demonstration that certain tasks cannot be performed without whole genome information, such as: identification of amino acid substitutions associated with clinical features (Simon et al.; 2019); analysis of quasispecies variants (Simon et al.; 2019)(Barbezange et al.; 2018); detection of reassortments (Simon et al.; 2019)(Nagarajan and Kingsford; 2010)(Goldstein et al.; 2018); detection of variants associated with vaccine status (Simon et al.; 2019); phylogenetic clustering of severe cases (Simon et al.; 2019)(Goldstein et al.; 2018). The superior resolution of WGS trees has been noted (Goldstein et al.; 2018). Other analyses emphasizing the superiority of WGS include a study performed by Meinel *et al.* (2018) (Meinel et al.; 2018b), who argued that WGS offered superior resolution to single-segment analysis; when analyzing clustering on the basis of single segments, the authors found that spurious clusters were identified.

Modern molecular epidemiology requires not only building a phylogenetic tree but, increasingly, divergence dating and phylodynamics as well. However, the gap between the former and the latter categories can be large. Arguably the most important software for the latter, BEAST, requires technical knowledge that exceeds the requirements for phylogeny construction, which has become routine, since models are more easy to miss-specify, and results of inference can be misleading or dangerous as a result. Furthermore, understanding of the data regimes for which these analyses are feasible, and how reliable they are, is not well understood. As such, there is considerable work required to improve the accessibility and standardization of these kinds of analyses. The ability to obtain good estimates for clock rates and tMRCA is dependent on many variables, such as the clock rate, number of samples, epidemic duration, and more. When the information in sequences is low, estimates can be poor (Guindon; 2010). Few studies benchmark performance of inference, and usually only for particular scenarios. For example, cross validation of tip dates is rarely used (Smith and OMeara; 2012; Sanderson; 2002). Some benchmarking studies do exist, such as (Duchêne et al.; 2016) and (Didelot et al.; 2018).

5.2.1 Objectives

The objectives of this chapter were to assess improvements in phylogenetic resolution for influenza virus molecular epidemiology using whole genome data, and demonstrate the data ranges in which downstream analyses become infeasible. Specifically, I aimed to:

1. Compare conventional phylogenetic resolution (such as node support) obtained when using conventional influenza HA/NA gene sequences to that of WGS data.

2. Assess whether BEAST can be utilized for dating WGS data for a single influenza season, and compare the results to those obtained with HA and NA alone.

3. Perform stochastic epidemic simulations in order to examine the theoretical data regimes for which molecular dating is performant, comparing BEAST (Bouckaert et al.; 2014), treedater (Volz and Frost; 2017), and treetime (Sagulenko et al.; 2018).

5.3 Methodology

5.3.1 Real datasets

Influenza A/H1N1pdm09 from the 2018/2019 season (155 samples) and A/H3N2 from both the 2017/2018 season (99 samples) and the 2018/2019 season (77 samples) were collected for both routine surveillance and clinical investigation. Only genome sequences with at least 90% coverage of HA, and 60% coverage of the other 7 segments, were included.

5.3.2 Comparison of resolution for increasingly complete genomes

In order to gain an indication of the extent to which the number of genome segments improves phylogenetic resolution, 77 influenza A/H3N2 whole genome sequences were used to build 8 phylogenetic trees, adding in segments for each: HA; HA-NA; HA-MP-NA; HA-MP-NA-NP; HA-MP-NA-NP-NS; HA-MP-NA-NP-NS-PA; HA-MP-NA-NP-NS-PA-PB1; HA-MP-NA-NP-NS-PA-PB1-PB2. Multiple sequence alignment was performed with MAFFT (Katoh et al.; 2009). Trees were built with iqtree (Nguyen et al.; 2015), with ModelFinder Plus for model selection, 1000 ultrafast bootstrap replicates (Hoang et al.; 2018), and rooted using Perth/16/2009 as an outgroup. In order to assess the accuracy of clades, the distribution of internal branch lengths, as well as bootstrap support as a function of number of genome segments, was recorded using BioPython (Hoang et al.; 2018). Trees were plotted using the R package ggtree (Yu et al.; 2017).

5.3.3 Simulation

For a demographic model, I use a general stochastic SIR model as previously described (Buckingham-Jeffery et al.; 2018) (Ball; 1986)(Allen; 2017). Briefly, let $\mathcal{S}_t, \mathcal{I}_t, \mathcal{R}_t$ be the sets of susceptible, infected, and recovered individuals at time t over the course of an outbreak, and $S_t = |\mathcal{S}_t|, I_t = |\mathcal{I}_t|, R_t = |\mathcal{R}_t|$ the counts in each class taking values in $\mathbb{N}_{\geq 0}$. I assume $\{(S_t, I_t), t \geq 0\}$, with $S_0 = N - I_0, I_0 = M$, and $S + I + R = N$ is a Markov process with transition probabilities as previously described. Trajectories were simulated with the Gillespie algorithm (Gillespie; 1977). Then, given a stochastic trajectory, an influenza genealogy was simulated by explicitly tracking a population evolving under this trajectory, assuming uniform sampling of participating infected individuals at each event. In turn, sequence evolution was simulated along the branches of these trees as with the Gillespie algorithm (analogously to above) under the assumption of a strict molecular clock with rate μ, and HKY model with parameter κ uniformly samples from $[0.5, 2.0]$, since nucleotide substitution rates can vary several fold (Posada and Crandall; 2001).

Since there is uncertainty regarding suitable parameter values $\theta = (N, M, R_0, \gamma, \mu)$, samples were drawn from $P(\theta, \{(S_t, I_t), t \geq 0\}) = P(\{(S_t, I_t), t \geq 0\}|\theta)P(\theta)$ as follows: firstly, parameters were drawn from prior distributions (as described in section *epidemiological priors*), followed by a trajectory conditional on θ, via the Gillespie algorithm (Gillespie; 1977). Simulation from a wide parameter space allows us to explore whether evolutionary analysis is sensitive to these parameter values. Because stochastic epidemics can die off quickly, rejection sampling was used (Hobolth and Stone; 2009) to condition on trajectories that result in sufficient infected individuals.

For the first simulated experiment, T1, transmission trees and sequence data were simulated from the whole concatenated genome of Perth/16/2009 (H3N2). Sample sizes were chosen to range between 10 and 50, with minimum epidemic size 1000, $S0 \sim \text{Uniform}(1000, 10000)$, $R0 \sim \text{Uniform}(1.1, 2.0)$, $\gamma = 90$ (per year, corresponding to an expected infectious duration of just over 4 days), $\mu \sim \text{Uniform}(0.0004, 0.004)$.

5.3.4 Molecular clock estimation

Treedater (Volz and Frost; 2017) was used for preliminary exploration of molecular clock signal by fitting a strict clock model (ω_0=0.003, searchRoot=nTips, maxit=5000), performing root-to-tip regression, calculation of quantile plots, and identification of outliers. Furthermore, in order to gain insight into the degree of molecular clock signal and accuracy in dating ancestral nodes in a single season with differing numbers of segments, this procedure was performed for both whole genomes for A/H1N1 and A/H3N2 datasets, and for A/H1N1 with each segment. Bayesian evolutionary modelling was performed with BEAST2 (Bouckaert et al.; 2014). For the purposes of assessing molecular clock estimation with differing genome coverage, for whole genome A/H3N2 (18/19) and HA-NA A/H3N2 (18/19) datasets, a flexible Bayesian skyride (Minin et al.; 2008), demographic prior with default priors was used with a HKY (Hasegawa et al.; 1985) substitution model (log Normal kappa (1.0, 1.25)), and strict molecular clock (Uniform prior (0.001, 0.004)). MCMC was run with a chain length of 10 million, sampling every 10,000 states.

5.3.5 Epidemiological priors

Estimates of infectious duration $(1/\gamma)$ vary, due to factors such as nonlinear dependence of infectivity on viral shedding. However, some estimates are presented in the range of 4-8 days (Tsang et al.; 2015). As described by Tsang *et al.* (2015), most transmissions occurs within a few days, with peak infectivity at approximately 1 day after the onset of symptoms. Although peak infectivity was around 1

day, this value is probably too small to use as a proxy for an exponential recovery rate. Other studies have estimated peak transmissability at day 2 (Carrat et al.; 2008). Furthermore, mean serial intervals have been estimated as 2.6 days for influenza (Suess et al.; 2010). In a small study of ferrets, the infectious duration was at least 3 days (Inagaki et al.; 2015), and for a few secondary infections 4 days, but none at 5 days. As such, a $1/\gamma \in [1, 5]$ is feasible. At the same time, estimates of peak viral shedding are around 2 days since innoculation (Carrat et al.; 2008). Without explicitly accounting for an exposed but not transmitting compartment E this extra day with an SEIR model, $1/\gamma$ was allowed to be higher than 1.

Estimates of seasonal influenza R_0 range from 0.9 to 2.1. This is lower than for novel pandemic strains (Coburn et al.; 2009). A similar review provided an IQR of 1.19-1.37 (Biggerstaff et al.; 2014). As such, the R_0 value was constrained to between 1.0 and 1.5, with a mean around 1.3. Again, due to the mass action assumptions of the model, some degree of variation around this value was allowed.

The clock rate of influenza is on the order of 10^{-4} to 10^{-3} per base per year (Shao et al.; 2017). Jang *et al.* (2018) estimated the substitution rate of H1N1 HA and NA to be around 2×10^{-3} and 1.8×10^{-3} respectively (Jang and Bae; 2018).

5.4 Results

5.4.1 WGS provides superior phylogenetic node support

Figures 5.1 and 5.2 show phylogenies for 77 influenza A/H3N2 samples collected during the 2018-2019 season, constructed with HA only and with all 8 segments. Phylogenetic resolution incrementally improved as segments were added into the reconstruction process (see also Figure 5.3). For example, consider the tree with all 8 segments. A group with 99% bootstrap support is observed, including taxa: A/Burry Port/1014/2019; A/Burry Port/8088/2019; A/Burry Port/1888/2019; A/Newport/1070/2019; A/Ebbw Vale/3447/2019. However, for the phylogenetic tree with HA alone, this group was not observed. Instead, these taxa are present within a larger polytomy consisting of a dozen taxa, with zero branch lengths. As another example, A/Lampeter/0442/2019 and A/Lampeter/7085 are grouped together with 98% bootstrap support on the whole genome tree, but not on the HA-only tree. Finally, the samples in the tree from Usk were not resolvable with HA alone; importantly, these were associated with an outbreak in 2019. Furthermore, boxplots in Figure 5.3 show the distribution of node supports for each tree. Mean node support increased as segments were added.

For the segment combinations as described, the mean internal node supports were: 69.27, 78.22, 77.6, 83.08, 86.09, 86.20, 88.45, 88.83. Interquartile ranges for each also increased dramatically. For HA alone, the lowest quartile was under 10%, and for the whole genome tree, was over 75%. These results demonstrate that for single seasons, influenza phylogenetic trees have increased resolution with whole genome sequences.

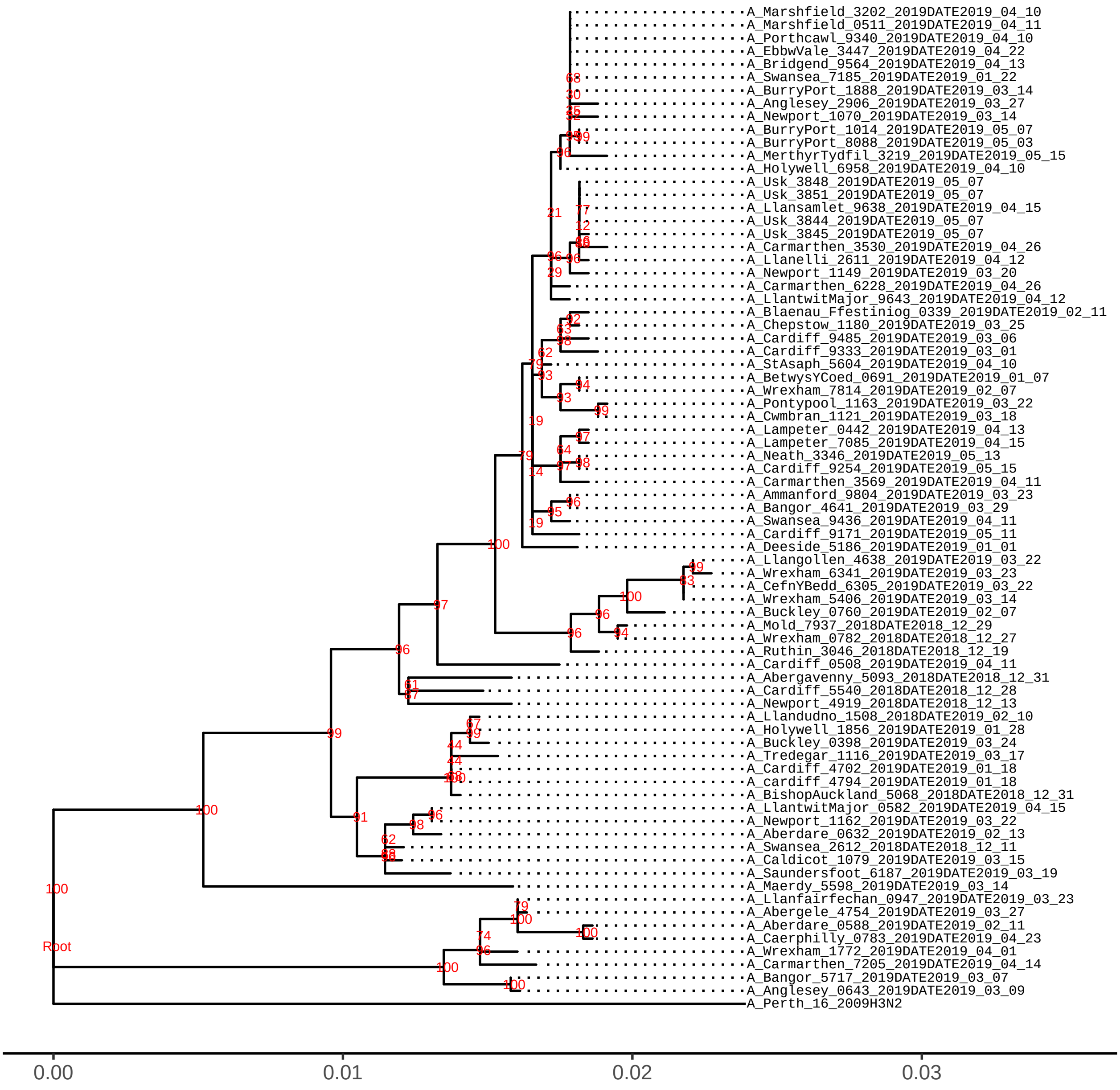

A_Marshfield_3202_2019DATE2019_04_10
A_Marshfield_0511_2019DATE2019_04_11
A_Porthcawl_9340_2019DATE2019_04_10
A_EbbwVale_3447_2019DATE2019_04_22
A_Bridgend_9564_2019DATE2019_04_13
A_Swansea_7185_2019DATE2019_01_22
A_BurryPort_1888_2019DATE2019_03_14
A_Anglesey_2906_2019DATE2019_03_27
A_Newport_1070_2019DATE2019_03_14
A_BurryPort_1014_2019DATE2019_05_07
A_BurryPort_8088_2019DATE2019_05_03
A_MerthyrTydfil_3219_2019DATE2019_05_15
A_Holywell_6958_2019DATE2019_04_10
A_Usk_3848_2019DATE2019_05_07
A_Usk_3851_2019DATE2019_05_07
A_Llansamlet_9638_2019DATE2019_04_15
A_Usk_3844_2019DATE2019_05_07
A_Usk_3845_2019DATE2019_05_07
A_Carmarthen_3530_2019DATE2019_04_26
A_Llanelli_2611_2019DATE2019_04_12
A_Newport_1149_2019DATE2019_03_20
A_Carmarthen_6228_2019DATE2019_04_26
A_LlantwitMajor_9643_2019DATE2019_04_12
A_Blaenau_Ffestiniog_0339_2019DATE2019_02_11
A_Chepstow_1180_2019DATE2019_03_25
A_Cardiff_9485_2019DATE2019_03_06
A_Cardiff_9333_2019DATE2019_03_01
A_StAsaph_5604_2019DATE2019_04_10
A_BetwysYCoed_0691_2019DATE2019_01_07
A_Wrexham_7814_2019DATE2019_02_07
A_Pontypool_1163_2019DATE2019_03_22
A_Cwmbran_1121_2019DATE2019_03_18
A_Lampeter_0442_2019DATE2019_04_13
A_Lampeter_7085_2019DATE2019_04_15
A_Neath_3346_2019DATE2019_05_13
A_Cardiff_9254_2019DATE2019_05_15
A_Carmarthen_3569_2019DATE2019_04_11
A_Ammanford_9804_2019DATE2019_03_23
A_Bangor_4641_2019DATE2019_03_29
A_Swansea_9436_2019DATE2019_04_11
A_Cardiff_9171_2019DATE2019_05_11
A_Deeside_5186_2019DATE2019_01_01
A_Llangollen_4638_2019DATE2019_03_22
A_Wrexham_6341_2019DATE2019_03_23
A_CefnYBedd_6305_2019DATE2019_03_22
A_Wrexham_5406_2019DATE2019_03_14
A_Buckley_0760_2019DATE2019_02_07
A_Mold_7937_2018DATE2018_12_29
A_Wrexham_0782_2018DATE2018_12_27
A_Ruthin_3046_2018DATE2018_12_19
A_Cardiff_0508_2019DATE2019_04_11
A_Abergavenny_5093_2018DATE2018_12_31
A_Cardiff_5540_2018DATE2018_12_28
A_Newport_4919_2018DATE2018_12_13
A_Llandudno_1508_2018DATE2019_02_10
A_Holywell_1856_2019DATE2019_01_28
A_Buckley_0398_2019DATE2019_03_24
A_Tredegar_1116_2019DATE2019_03_17
A_Cardiff_4702_2019DATE2019_01_18
A_cardiff_4794_2019DATE2019_01_18
A_BishopAuckland_5068_2018DATE2018_12_31
A_LlantwitMajor_0582_2019DATE2019_04_15
A_Newport_1162_2019DATE2019_03_22
A_Aberdare_0632_2019DATE2019_02_13
A_Swansea_2612_2018DATE2018_12_11
A_Caldicot_1079_2019DATE2019_03_15
A_Saundersfoot_6187_2019DATE2019_03_19
A_Maerdy_5598_2019DATE2019_03_14
A_Llanfairfechan_0947_2019DATE2019_03_23
A_Abergele_4754_2019DATE2019_03_27
A_Aberdare_0588_2019DATE2019_02_11
A_Caerphilly_0783_2019DATE2019_04_23
A_Wrexham_1772_2019DATE2019_04_01
A_Carmarthen_7205_2019DATE2019_04_14
A_Bangor_5717_2019DATE2019_03_07
A_Anglesey_0643_2019DATE2019_03_09
A_Perth_16_2009H3N2
Root
0.00
0.01
0.02
0.03

Figure 5.1: **Phylogenetic tree for 77 H3N2 HA sequences from the 2018-2019 season**. Several polytomies occur toward the top of the tree. Importantly, samples associated with Usk cannot be resolved. Red numbers indicate bootstrap suport.

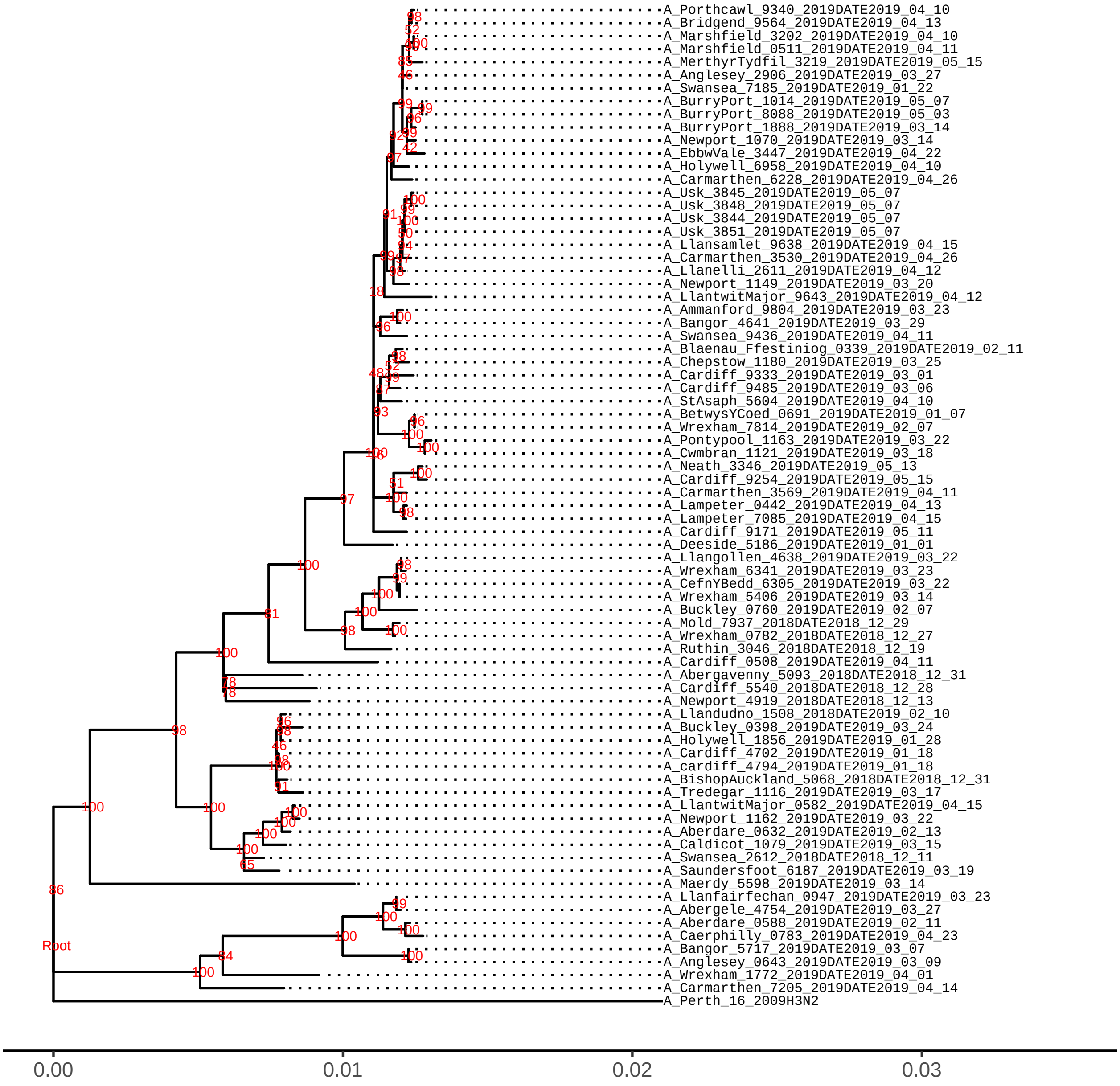

A_Porthcawl_9340_2019DATE2019_04_10
A_Bridgend_9564_2019DATE2019_04_13
A_Marshfield_3202_2019DATE2019_04_10
A_Marshfield_0511_2019DATE2019_04_11
A_MerthyrTydfil_3219_2019DATE2019_05_15
A_Anglesey_2906_2019DATE2019_03_27
A_Swansea_7185_2019DATE2019_01_22
A_BurryPort_1014_2019DATE2019_05_07
A_BurryPort_8088_2019DATE2019_05_03
A_BurryPort_1888_2019DATE2019_03_14
A_Newport_1070_2019DATE2019_03_14
A_EbbwVale_3447_2019DATE2019_04_22
A_Holywell_6958_2019DATE2019_04_10
A_Carmarthen_6228_2019DATE2019_04_26
A_Usk_3845_2019DATE2019_05_07
A_Usk_3848_2019DATE2019_05_07
A_Usk_3844_2019DATE2019_05_07
A_Usk_3851_2019DATE2019_05_07
A_Llansamlet_9638_2019DATE2019_04_15
A_Carmarthen_3530_2019DATE2019_04_26
A_Llanelli_2611_2019DATE2019_04_12
A_Newport_1149_2019DATE2019_03_20
A_LlantwitMajor_9643_2019DATE2019_04_12
A_Ammanford_9804_2019DATE2019_03_23
A_Bangor_4641_2019DATE2019_03_29
A_Swansea_9436_2019DATE2019_04_11
A_Blaenau_Ffestiniog_0339_2019DATE2019_02_11
A_Chepstow_1180_2019DATE2019_03_25
A_Cardiff_9333_2019DATE2019_03_01
A_Cardiff_9485_2019DATE2019_03_06
A_StAsaph_5604_2019DATE2019_04_10
A_BetwysYCoed_0691_2019DATE2019_01_07
A_Wrexham_7814_2019DATE2019_02_07
A_Pontypool_1163_2019DATE2019_03_22
A_Cwmbran_1121_2019DATE2019_03_18
A_Neath_3346_2019DATE2019_05_13
A_Cardiff_9254_2019DATE2019_05_15
A_Carmarthen_3569_2019DATE2019_04_11
A_Lampeter_0442_2019DATE2019_04_13
A_Lampeter_7085_2019DATE2019_04_15
A_Cardiff_9171_2019DATE2019_05_11
A_Deeside_5186_2019DATE2019_01_01
A_Llangollen_4638_2019DATE2019_03_22
A_Wrexham_6341_2019DATE2019_03_23
A_CefnYBedd_6305_2019DATE2019_03_22
A_Wrexham_5406_2019DATE2019_03_14
A_Buckley_0760_2019DATE2019_02_07
A_Mold_7937_2018DATE2018_12_29
A_Wrexham_0782_2018DATE2018_12_27
A_Ruthin_3046_2018DATE2018_12_19
A_Cardiff_0508_2019DATE2019_04_11
A_Abergavenny_5093_2018DATE2018_12_31
A_Cardiff_5540_2018DATE2018_12_28
A_Newport_4919_2018DATE2018_12_13
A_Llandudno_1508_2018DATE2019_02_10
A_Buckley_0398_2019DATE2019_03_24
A_Holywell_1856_2019DATE2019_01_28
A_Cardiff_4702_2019DATE2019_01_18
A_cardiff_4794_2019DATE2019_01_18
A_BishopAuckland_5068_2018DATE2018_12_31
A_Tredegar_1116_2019DATE2019_03_17
A_LlantwitMajor_0582_2019DATE2019_04_15
A_Newport_1162_2019DATE2019_03_22
A_Aberdare_0632_2019DATE2019_02_13
A_Caldicot_1079_2019DATE2019_03_15
A_Swansea_2612_2018DATE2018_12_11
A_Saundersfoot_6187_2019DATE2019_03_19
A_Maerdy_5598_2019DATE2019_03_14
A_Llanfairfechan_0947_2019DATE2019_03_23
A_Abergele_4754_2019DATE2019_03_27
A_Aberdare_0588_2019DATE2019_02_11
A_Caerphilly_0783_2019DATE2019_04_23
A_Bangor_5717_2019DATE2019_03_07
A_Anglesey_0643_2019DATE2019_03_09
A_Wrexham_1772_2019DATE2019_04_01
A_Carmarthen_7205_2019DATE2019_04_14
A_Perth_16_2009H3N2
Root
86
100
98
100
100
100
97
100
0.00
0.01
0.02
0.03

Figure 5.2: **Phylogenetic tree for 77 H3N2 whole genome sequences from the 2018-2019 season.** Most of the internal nodes for Usk outbreak subclades have high bootstrap support (indicated by red numbers).

5.4.2 WGS can provide sufficient information for molecular clock estimation for a single influenza season

Molecular clock estimation and diagnostics were performed with treedater (Volz and Frost; 2017) for all three WGS datasets (A/H1N1 18/19, A/H3N2 17/18, A/H3N2 18/19). In all cases, a strict molecular clock was estimated. For A/H1N1 (18/19), the estimated clock rate was 2.50×10^{-3} and tMRCA was 2.26 years. For A/H3N2 (17/18), the rate was estimated as 3.05×10^{-3} and tMRCA 2.64 years. For A/H3N2 (18/19), the rate was estimated as 3.16×10^{-3}, with a tMRCA of 3.21 years. It should be noted that given the datasets, the results are exploratory and indicative of clock signal, but bias

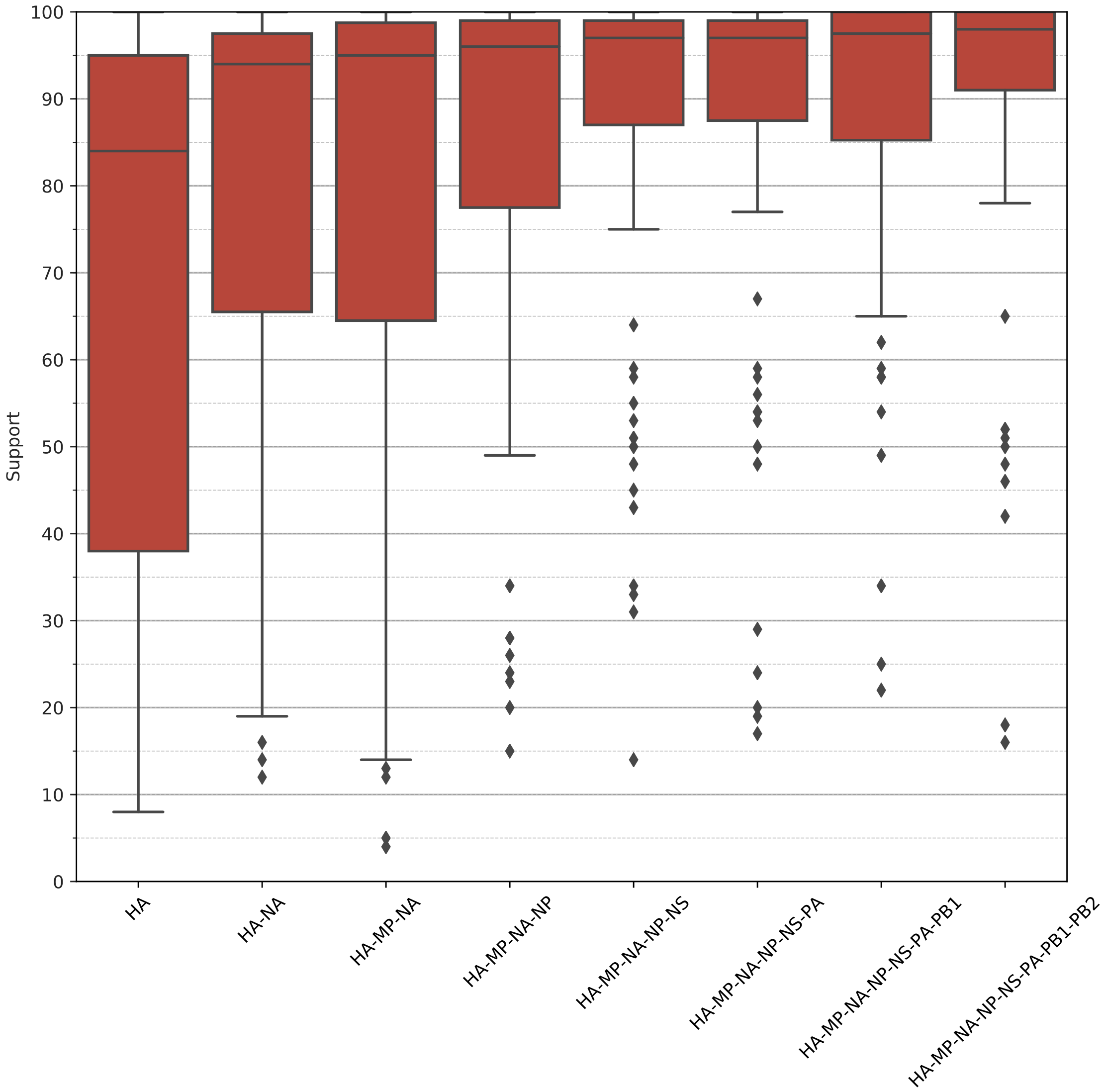


Figure 5.3: **Boxplots showing node supports for increasing number of genome segments.** Node supports improve with increasing number of segments, up to around 5 segments. For HA alone, the bootstrap support of nodes is quite poor.

is introduced in both the tree-building procedure, population structure, rate variation, and possible reassortment. Given that, within each tree, disparate lineages that diverged several years ago exist, it is probable that factors of this kind biased clock-rate estimates. This can be observed in the root-to-tip regression plots, and quantile plots, where deviations are observed. As a comparison, using only the HA and NA genes of the A/H3N2 (18/19) dataset, the clock-rate was estimated to be 2.8×10^{-3}, with a tMRCA of 5.82 years. Compared with the whole genome dataset with RTT regression ($p = 6.05 \times 10^{-5}$), the RTT regression was still significant $p = 0.0129$. Although the clock rates estimated from the two datasets was similar, the TMRCA was nearly twice as old.

As shown in Figure 5.4, inference on HA and NA alone with BEAST yielded $\hat{\mu}$ with a much higher variance. Figure 5.4 gives posterior distributions for strict molecular clock with both whole genome and HA and NA alone. For HA and NA, the posterior mean estimate was 4.45×10^{-3}, with a 95% HPD of $[3.0 \times 10^{-3}, 5.9 \times 10^{-3}]$. However, for the whole-genome sequences, the posterior mean was 2.45×10^{-3}, with a 95% HPD interval of $[1.49 \times 10^{-3}, 3.51 \times 10^{-3}]$. This result indicates that, within a single season, influenza whole genome sequences offer more precise molecular clock estimates. However, since μ is unknown (and in fact likely to have high variance), it is not known which mean was closer to the true value. It should be added that, even with HA and NA only, the molecular clock estimate 95% HPD did only span about 5.9×10^{-3}, which could be considered reasonable.

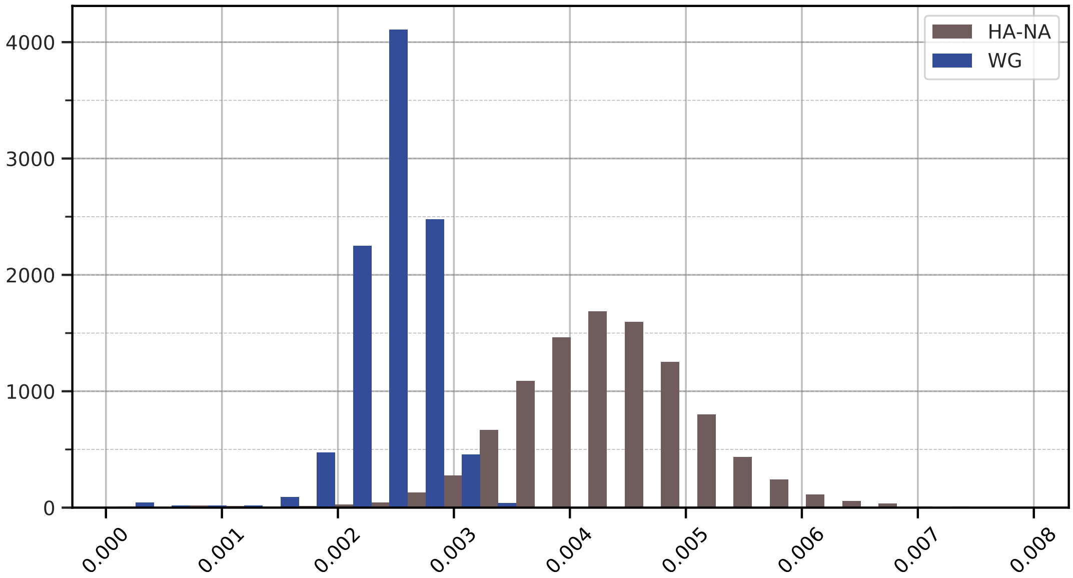


Figure 5.4: **Posterior samples for μ for HA and NA alone (grey) and whole genome (blue) for 77 H3N2 samples from the 2018-2019 season.** The estimates derived from whole genomes had lower variance than those from only HA and NA.

5.4.3 Simulations

For simulated populations, the mean epidemic duration was 0.314 years (standard deviation 0.111), mean diversity was 5.423 (standard deviation 3.350), mean total number of infected individuals was 3177.113 (standard deviation 1827.73). Figure 5.5 summarizes the statistics of these simulations.

All methods performed reasonably well in simulation, even on an average time scale of 3.6 months, as shown in Figure 5.6. The mean normalized error $(\mu - \hat{\mu})/\mu$ for treedater, treetime, BEAST-HKY, BEAST-JC, ad BEAST-HKY-VAL were 0.309, 0.032, -0.007, 0.0047, and 0.0008. As demonstrated by fixing the tree with BEAST-HKY-VAL, the clock rate estimation was the best. Furthermore, the difference between BEAST-HKY and BEAST-JC was small, implying that the substitution model itself was not as important for clock rate estimation, at least for κ in the regime tested. For BEAST-HKY and BEAST-JC, I assessed the proportion of node times that fell into the estimated 95% HPD interval, which were 0.849 and 0.852, respectively. This indicated that most of the time, for these simulations, TMRCA dates fell within the margin of error (this was not 95%, but some error is expected, since the model is not exactly specified, by design). TMRCAs were also estimated, as shown in Figure 5.7. Errors were normalized by epidemic duration, since for longer epidemics, ($(T - \hat{T})/H$, where H is the epidemic duration, since we are interested in the error in terms of the epidemic timescale). The mean errors were 1.340, 0.568, 0.354, 0.375, and 0 for treedater, treetime, BEAST-HKY, BEAST-JC, and BEAST-HKY-VAL (the last had a fixed tree). As such, the best performing, BEAST-HKY, had a mean error in TMRCA of about 35% of the epidemic duration. These results imply that high-precision estimation of TMRCAs may still be difficult.

5.5 Discussion

5.5.1 Resolution

Phylogenetic reconstruction was found to be more confident for WGS (that is, each node had higher bootstrap support, and polytomies were more resolved) during our analysis, and bootstrap support across nodes increased as a function of the number of segments. However, as is the case with many phylogenetic analyses, it cannot be stated that these reconstructions were in fact more *accurate*, because it is not possible to validate the true phylogenetic relationship between the samples. Although it may seem that the HA-only tree exhibited the main structural features of the whole genome tree, the tree itself represents multiple introductions into Wales (that is, the virus was imported multiple times). Further analysis (see Chapter 7) can be used to examine which subclades may represent imported lineages. In principle, I note that 5 segments may be sufficient; however, this many segments are only sequenced typically by whole genome sequencing.

Importantly, the samples associated with the Usk 2019 outbreak were only resolved on the whole genome tree. This indicates that, for routine epidemiological investigation, if analyses are required on the level of outbreaks, the whole genome is required. For example, in order to assess whether a community sample sits within an outbreak (implying outbreak to community transmission), those samples must be resolved.

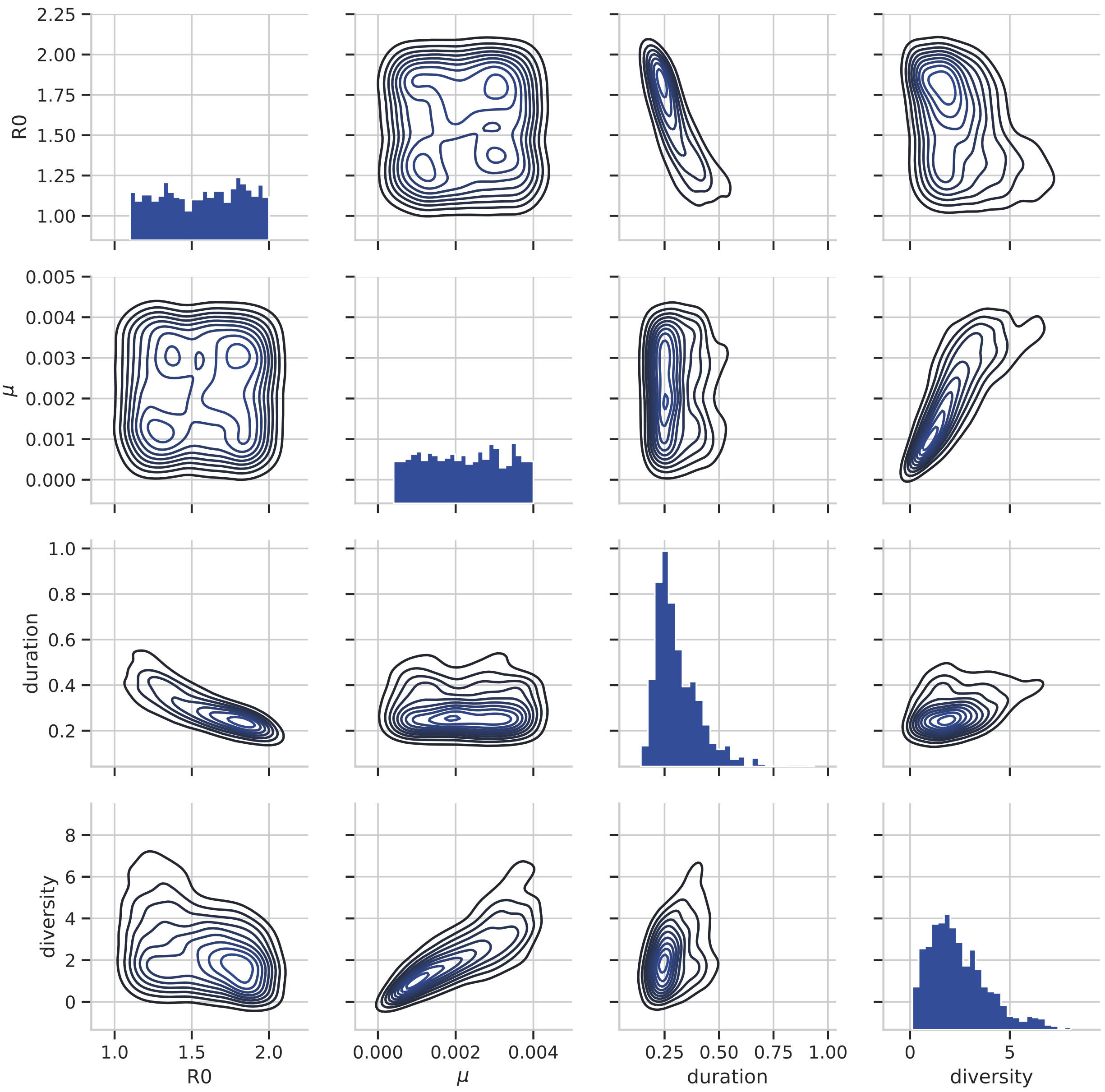


Figure 5.5: **Basic statistics for simulated epidemics.** Some parameters are expectedly associated, such as diversity and μ, or epidemic duration and R_0.

5.5.2 Molecular clock estimation

For molecular clock estimation, our results suggest that the whole genome offers the best performance. Using BEAST for real data, it was found that estimates using the whole genome versus those with just HA and NA resulted in much smaller 95% credible intervals. The posterior distribution using the whole genome was more consistent with estimates of μ from the literature (Jang and Bae; 2018). The posterior mean for HA and NA, at 4.45×10^{-3}, was likely too high. However, it should be

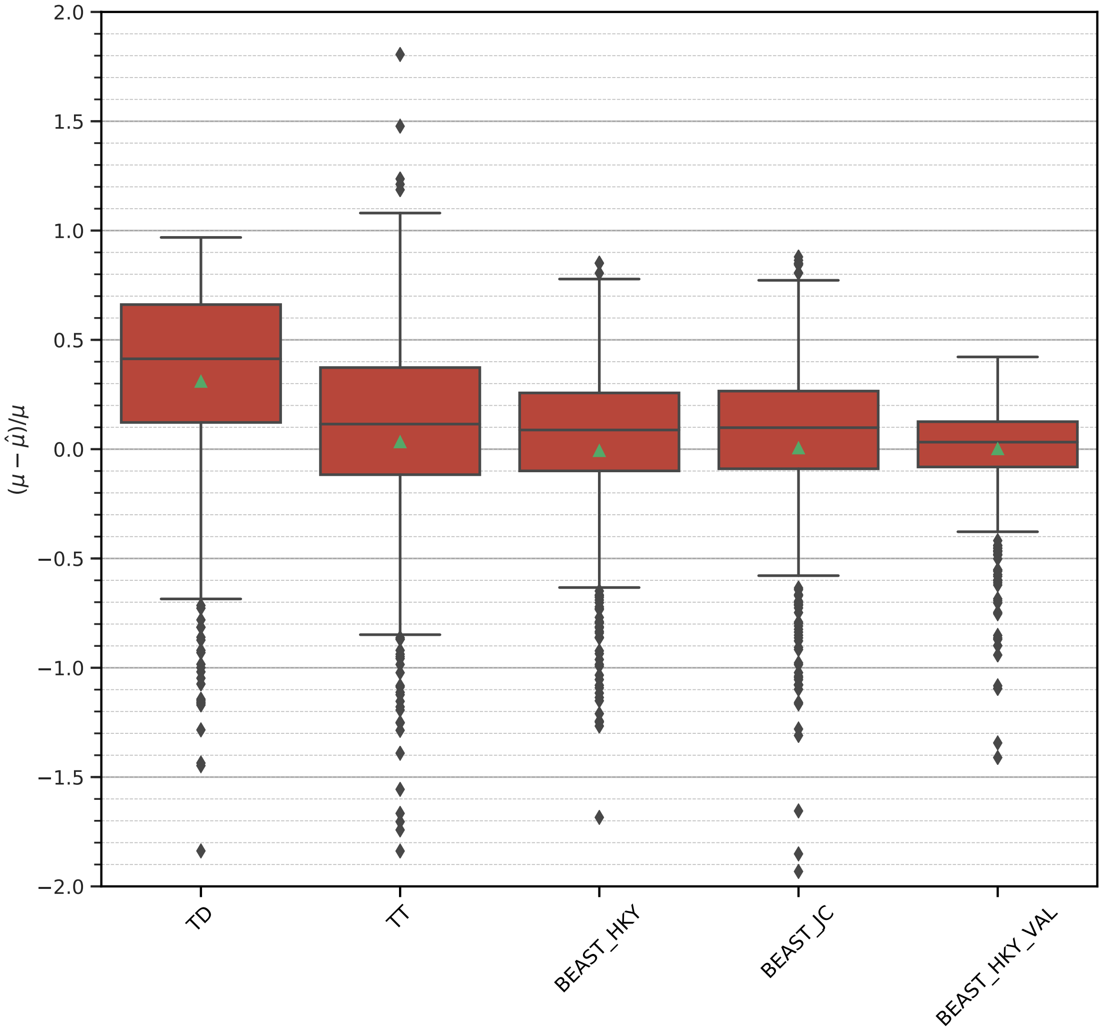


Figure 5.6: **Box-plots giving normalized error in clock rate for each method**. BEAST performed best of these methods, though there was not a large difference between HKY and JC models. For the fixed tree, BEAST-HKY-VAL, some error was still observed, implying that the tree building process does not drive all or most of the resultant uncertainty.

reiterated that it is not possible to prove that the whole genome estimates were more accurate than those from HA and NA alone. I note that, using treedater, the clockrate estimates were reasonable for both datasets. However, the TMRCAs (for example A/H3N2 18/19) were nearly 2 times larger with HA and NA alone. I therefore caution results obtained using only HA an NA.

For our simulations, BEAST performed the best, and HKY and JC models performed approximately the same, despite the fact that simulations were performed according to the HKY model. This indi-

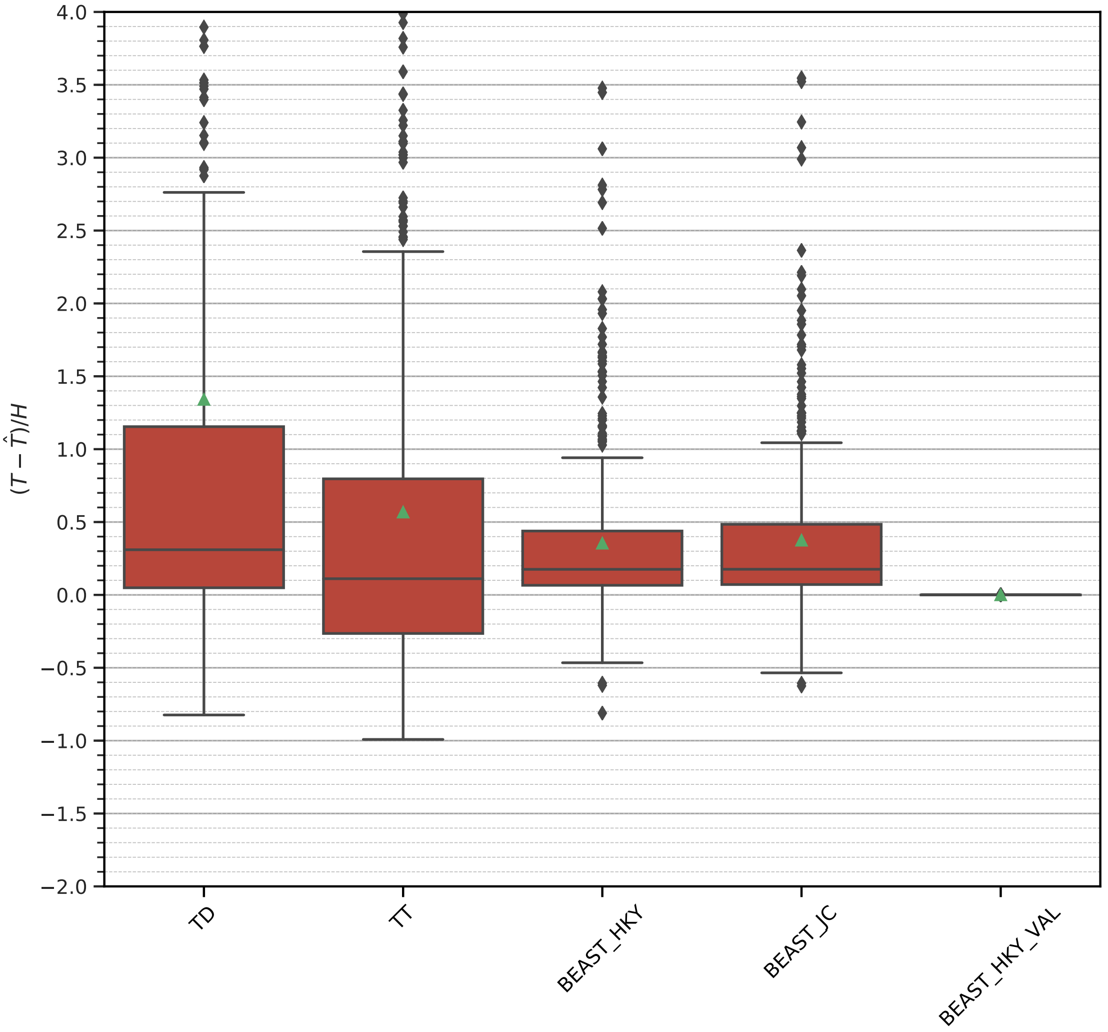


Figure 5.7: **Box-plots giving normalized error in tMRCA for each method**. The error $(T - \hat{T})$ is normalized by epidemic duration H, in order to allow comparison of simulated epidemics with variable durations (sometimes several times). In the best case, the mean error in tMRCA was around 25% of the epidemic duration.

cated that, at least for our epidemic time scales, the exact substituion model was not crucial. In all, the clock-rate estimates were reasonably good for these simulations; for BEAST and the fixed tree, the posterior mean was less than 1% different from the true value. Treedater was found to perform worse, with an average of 30% error in the clock rate.

Conversely, it was found that estimation of TMRCA was worse for all methods. Even for BEAST, this could be frequently incorrect by over 30% of the epidemic duration. In all cases, inference appeared to be biased. Further research could be done to elucidate the causes of this bias in TMRCA. Until further study can be performed to better understand the computational or statistical reasons for this, I suggest that in routine epidemiology, estimation of the TMRCA should be performed with cautious interpretation.

I note that, although filters were applied, outliers could have resulted from computational problems, such as poor ESS for BEAST, or an insufficient number of iterations for treedater.

5.5.3 Tree Prior

It is noted that previously, general epidemic models have been utilized in Bayesian inference (Vaughan et al.; 2019), though inference is much slower. Furthermore, epidemiological parameter values in this case were not a subject of interest. The tree prior that was chosen to use for BEAST, the flexible Bayesian skyride, can capture a range of dynamics, but due to the discrete nature of the process (effective population size can vary in time intervals, which themselves have a Brownian motion prior), it is not exact.

5.5.4 Simulations

Whilst these simulations, using the general stochastic epidemics, were more realistic than ODEs, they were not realistic. In particular, real datasets for epidemic viruses feature considerable structure due to importation, or population variables such as age. These simulations were, in a sense, reflective of the simplest possible epidemic scenarios. In that sense, the performance on these simulated datasets may be the best possible.

Chapter 6

Exponential growth modelling of SARS-CoV-2 during the first pandemic wave with phylodynamics

6.1 Abstract

Phylodynamic modelling has seen increasing use for estimation of epidemiological and evolutionary parameters of the COVID-19 pandemic. Here, a coalescent exponential growth model was employed to estimate growth rates, doubling time, and prevalence of COVID-19 prior to 23/03/2020 in the Cardiff postcode area in Wales. 42 publicly available SARS-CoV-2 whole genome sequences collected by Public Health Wales (PHW) were used. Doubling time was found to be 7.12 days (95% HPD: 3.68 to 25.58), which is consistent with other reports. Prevalence was estimated to be 30,599 (95% HPD: 801 to 222,648), which corresponded to approximately 3% (95% HPD: 0.07 to 22%) of the local population, the mean of which is likely to be an overestimate. Importantly, I demonstrate that prevalence estimates are sensitive to several epidemiological point estimates. These, and similar results, should be interpreted with care, and referred to in comparison to the results of other modelling approaches.

6.2 Introduction

Phylodynamic models have begun to demonstrate utility in estimation of the evolutionary and epidemiological parameters of pandemic viruses from sequence data. These approaches have previously been used in the study of the H1N1 pandemic in 2009, and considered beside other modelling approaches (Fraser et al.; 2009). Recent examples applied to the COVID-19 pandemic include estimation of molecular clock rates (Rambaut; 2020), phylogeography (Lycett; 2020), exponential growth models (Danesh et al.; 2020), estimation of R_0 with birth-death models (Vaughan et al.; 2020), and estimation of both incidence and prevalence (Bedford; 2020). As another example, refer to the preliminary report by Volz *et al.* (2020), where parameters of exponential and SEIR models were estimated with sequence data (Volz et al.; 2020).

Here I employ a coalescent model with exponential growth demographic (see (Kuhner et al.; 1998; Drummond et al.; 2002)), as well as point estimates of epidemiologicial parameters from literature and recent reports, to estimate prevalence of COVID-19 in the Cardiff postcode area, before 23/03/2020. For examples of exponential growth models applied to phylodynamic inference see (Faria et al.; 2014; Shiino et al.; 2010; Salemi et al.; 2008).

6.3 Methods

6.3.1 Sampling

Sampling bias, such as epidemiological linkage or geographical biases, can effect parameter estimation for phylodynamics (de Silva, Ferguson and Fraser; 2012), as can sampling strategies (Hall et al.; 2016; Frost and Volz; 2010), such as temporal distribution (Stack et al.; 2010), and sub-sampling (which may also be required for computational reasons). As such, 42 samples collected earlier than 23/04/20 (before lock-down) near Cardiff (identified by a CF outer postcode) were used, although epidemiological linkage of these samples was not known. Also, although Kingman's coalescent (Kingman; 1982) assumes a small sample fraction, it has been shown to be fairly robust to violations of this assumption (Fu; 2006). As such, for samples taken in South Wales, it should be noted that sampling fraction may be higher than normal, though it is not believed that this fraction would be high enough to substantially effect parameter estimates.

6.3.2 Investigation of temporal signal

A preliminary tree (HKY+G) was constructed with IQ-TREE (Nguyen et al.; 2015), and root-to-tip regression with best-fitting root was performed with TempEst (Rambaut et al.; 2016).

6.3.3 Model

A HKY DNA substitution model was used with a strict molecular clock (Hasegawa et al.; 1985), and a coalescent exponential growth demographic model. Models were constructed and run with the BEAST2 software (Bouckaert et al.; 2014) with a chain length of 10 million, and samples logged every 1000 states.

6.3.4 Priors

Previous estimates of the molecular clock rate, μ, have varied. However, a previous analysis with global sequences gave a 95% highest posterior density (HPD) interval of 0.14×10^{-3} to 1.31×10^{-3}, with a posterior mean of 0.8×10^{-3} (Rambaut; 2020). Given that several analyses have been produced independently (although perhaps with overlapping datasets) with values in a similar range, an informative log-normal prior was used for clock rate, ('real space' mean 9.0×10^{-4}; stdev 0.1). Uniform priors were used for exponential growth rate ($r \in [0, 1000]$) and initial effective population size, $N_e(0) \in [0, 1000]$). For substitution model prior κ, a default log-normal (mean 1.0, stdev 1.25) was used.

6.3.5 Prevalence calculation

When employing the coalescent model, the quantity N_e is inferred, rather than the true population size. N_e is a quantity that is, in the presence of offspring distributions with higher variance than that of the Poisson distribution (Fraser and Li; 2017), lower than the true population size. τ is the generation time. As performed and suggested by Volz *et al.* (Bedford; 2020; Volz et al.; 2020, 2013), as well as Bedford (Bedford; 2020), I utilized the equation derived by Fraser and Li (2017) (Fraser and Li; 2017):

$$N_e(t) = \frac{N(t)}{\sigma^2/R + R - 1}$$

where R and σ^2 are reproduction number and variance of the offspring distribution, respectively. As performed by Bedford (Bedford; 2020), an R_0 of 2.68 (Wu et al.; 2020) was used with a generation time of 7.5 days (Li et al.; 2020). Furthermore, it was assumed that the offspring distribution (here a negative binomial is assumed) is overdispersed, based on previous studies of SARS (Lloyd-Smith et al.; 2005) ($\hat{k} = 0.16$), as well as a recent report (in press) from the LSHTM ($\hat{k} = 0.1$) (Endo; 2020). Parameterized in terms of the mean R and dispersion, k, the variance of the NB is $R(1+R/k)$ (Lloyd-Smith; 2007). Finally, a generation time $\tau = 7.5$ days was used.

6.4 Results

Exponential growth rate had a posterior mean of 35.51 and 95% HPD interval of $[9.89, 68.83]$ (ESS: 507). This growth rate is similar to that previously estimated (Bedford; 2020), corresponding to an estimated doubling time of 7.12 days (95% HPD: $[3.68, 25.58]$), which is also similar to previous epidemiological estimates from Wuhan (7.4 days) (Li et al.; 2020). Effective population size had a posterior mean of 7.5, and 95% HPD interval of $[0.5, 49.9]$ (ESS: 583). The molecular clock-rate was estimated to be 8.37×10^{-4} (95% HPD interval: $[6.82 \times 10^{-4}, 9.91 \times 10^{-4}]$; ESS: 3958), and time to most recent common ancestor (TMRCA) of 69.02 (95% HPD interval: $[32.485, 145.3795]$; ESS 408) days. Figures 6.1 and 6.2 give estimated trajectories of both effective population size, $N_e(t)$, and prevalence, $I(t)$. These results suggest that, on 22/03/20, an estimated 30599.33 individuals were infected (95% HPD: $[801.82, 222648.03]$). Given an approximate population size estimate of 1005334 for this area (ONS, 2015) (ONS; 2020), an approximate fraction of 3% is estimated to have been infected (95% HPD: 0.07 to 22%). Finally, in order to gauge the sensitivity of these estimates to the assumed dispersion parameter k, Figure 6.3 gives the estimated prevalence on 22/03/20 with different values. As shown, for values less than 0.1, estimates of prevalence can increase sharply, so point estimates are unlikely to be suitable.

6.5 Conclusions

As noted in a comment by du Plessis (Rambaut; 2020), early estimates of clock-rate are difficult. Here, I used a strong prior on clock-rate, so estimates should additionally be interpreted with caution. I reiterate that these concerns also apply, if not more strongly, to demographic inference. The results presented in particular suffer from: i) sampling biases, in particular possible epidemiological linkage, as well as sequences sampled only from hospitals ii) poor fit of exponential model beyond the start of the outbreak, where growth is slower than exponential iii) use of point estimates of parameters such as k, to which results are clearly sensitive. These results should therefore be interepreted with caution in comparison to other modelling studies.

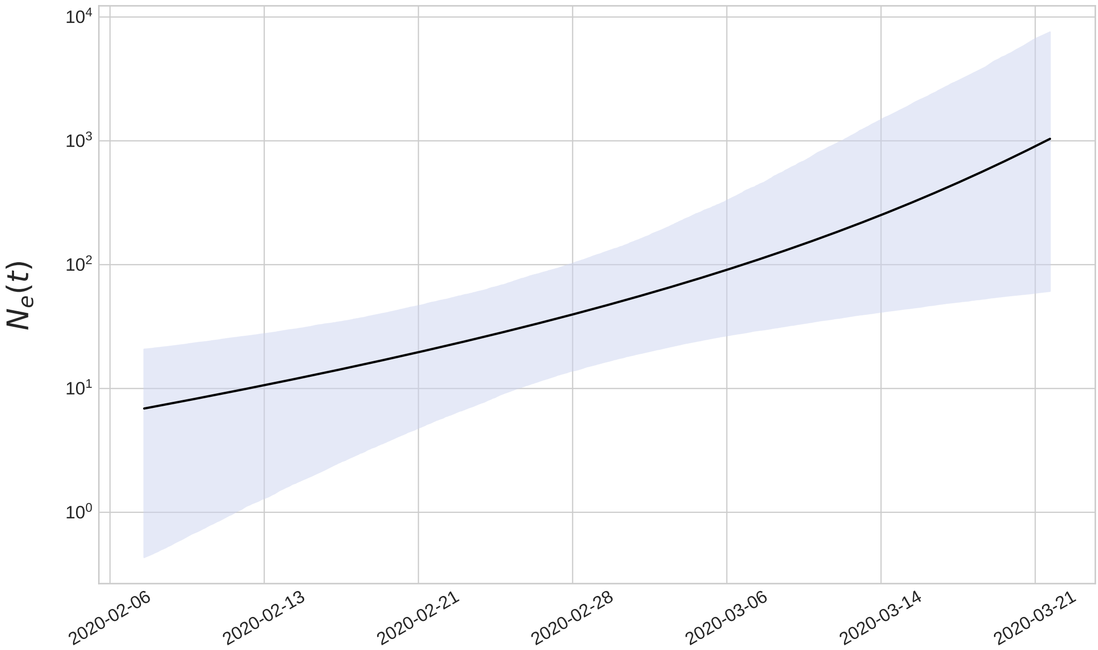


Figure 6.1: **Estimated effective population size over time in the Cardiff postcode area before lockdown.** Black line: posterior mean. Shaded area: 95% HPD interval.

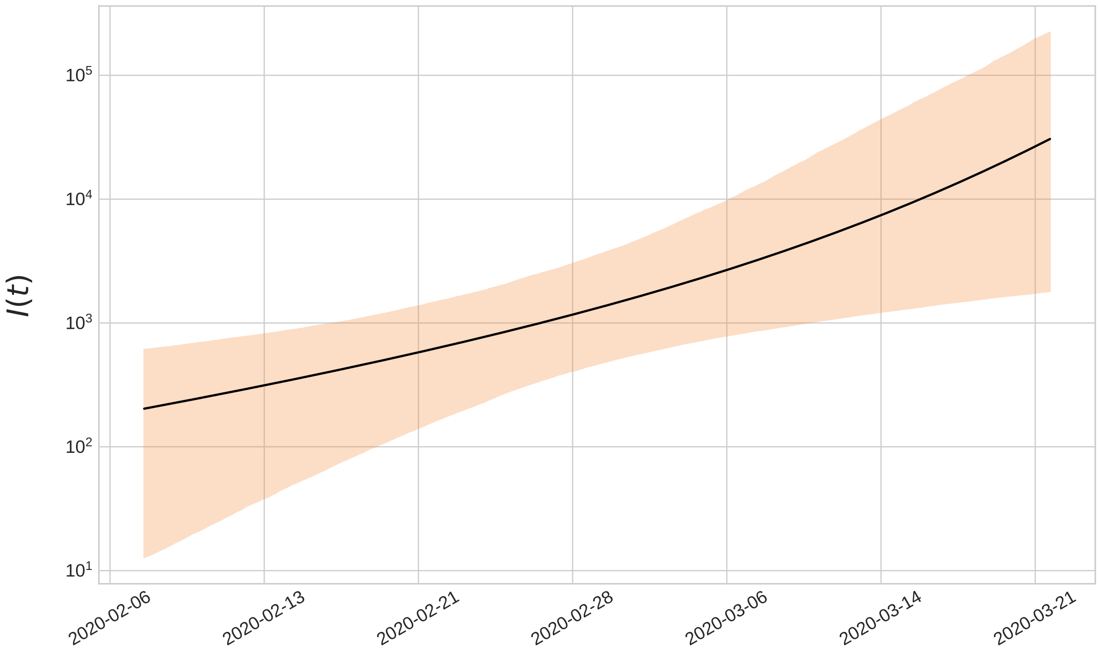


Figure 6.2: **Estimated number of infected individuals over time in the Cardiff postcode area before lockdown.** Black line: posterior mean. Shaded area: 95% HPD interval.

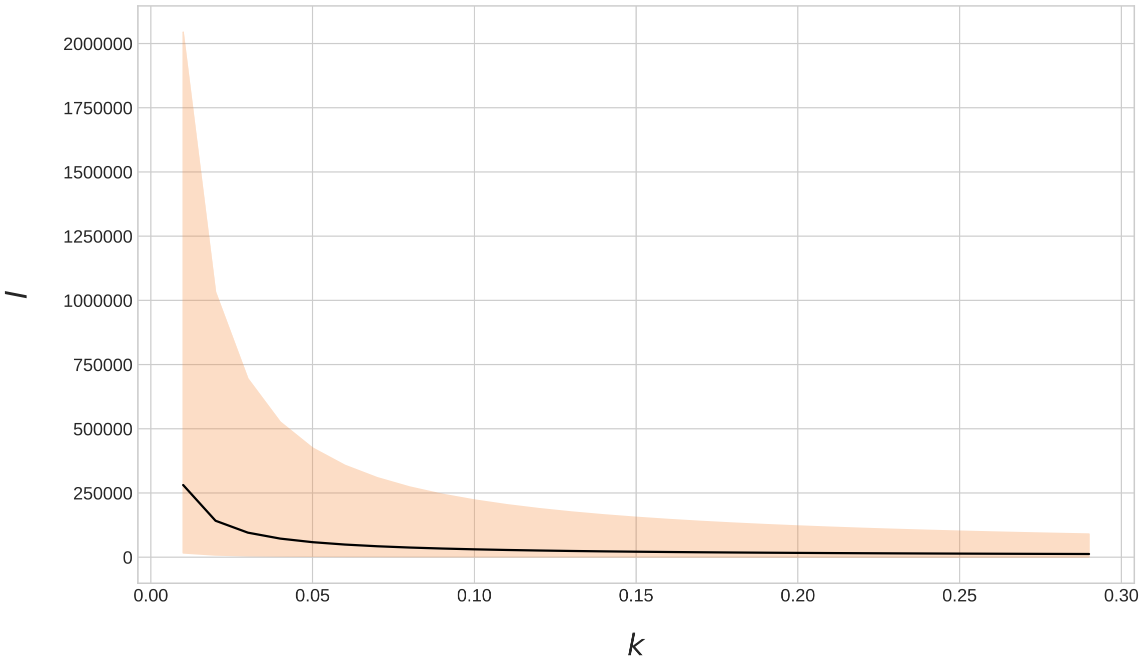


Figure 6.3: **Sensitivity of prevalence estimate on 22/03/20 to the assumed dispersion parameter k of the offspring distribution.** Black line: posterior mean. Shaded area: 95% HPD interval.

Chapter 7

Characterizing the importation of respiratory viruses with phylogenetic methods

7.1 Abstract

7.1.1 Background

In Wales, whole-genome sequencing (WGS) was rapidly deployed during the COVID-19 pandemic to help characterize the genome of SARS-CoV-2, utilizing many of the techniques first developed for epidemiological surveillance of influenza. Genomic data is unique in that it offers potential insight into the patterns of importation of a virus into, and within, a single country. Furthermore, once importation has been characterized, epidemiological cluster investigations can be simpler, since different imported sub-lineages can be excluded as epidemiologically unrelated. Here, I aimed to demonstrate applicability of these methods for both influenza and SARS-CoV-2 data.

7.1.2 Methods

Here, ancestral state reconstruction methods were applied to 77 influenza A/H3N2 samples from across Wales to examine population structure in relation to importation on a micro-scale. Then, utilizing 6,243 publicly available SARS-CoV-2 genomes sequenced by Public Health Wales as part of the COVID-19 pandemic response, the macro-scale trends in the rate of importation throughout the pandemic were characterized.

7.1.3 Results

For the influenza A/H3N2 dataset, characterization of imported sub-lineages allowed for the clarification of the relationships between samples. Furthermore, for the COVID-19 dataset, results indicated a trend in the rate of importation throughout the pandemic, beginning with a decrease from the early months, followed by a peak coinciding with the relaxation of lockdown, shortly before the second wave.

7.2 Introduction

Phylogenetic methods have found routine use in cluster investigation (Poon et al.; 2016; Kim et al.; 2017; Roy et al.; 2019; Lau et al.; 2016). However, due to the large quantities of genomic data available for many viruses, phylogenetic methods can now be applied to the examination of both spatio-temporal dynamics, and, pertinently, importation and exportation between nations (Dellicour et al.; 2020; da Silva Candido et al.; 2020; Dudas et al.; 2017; Baillie et al.; 2011; Bahl et al.; 2011). It has been demonstrated for many viruses that throughout an epidemic, repeated importation and extinction occurs. For example, multiple importation events of Zika virus into the United States were characterized (Grubaugh et al.; 2017) using molecular dating methods, as well as incidence and traffic data. A pre-print by du Plessis *et al.* (2020) (du Plessis et al.; 2020) examined the importation dynamics of SARS-CoV-2 during the first wave of the pandemic. Importantly, their study made use of phylogenetic methods augmented with travel data. Many of these methods make use of phylogeography, in particular those implemented in BEAST (Lemey et al.; 2009; Baele et al.; 2018). Lemey *et al.* (2014) (Lemey et al.; 2014) used discrete phylogography to combine mobility data with gene sequence data and demonstrate the impact of air travel on the dynamics of H3N2. In this study, the authors quantify the amount of "trunk time" that is taken up by each discrete location. Similarly, (Bedford et al.; 2010) define locations that dominate the evolutionary tree trunk of influenza A (H3N2). Lemey *et al.* (2020) (Lemey et al.; 2020) adapted these methods to incorporate travel histories. Interestingly, they demonstrate the Markov jump trajectory plot. Due to the rapid accumulation of sequence data during the COVID-19 pandemic, efforts have also been directed toward the development of analytical pipelines (Dellicour et al.; 2020); Dellicour *et al.* (2020), in their pre-print, demonstrate their pipeline on sequence data from Belgium, and point out that Bayesian methods may be too slow for large datasets. 'Markov jumps' (Minin and Suchard; 2008a,b), which represent transitions between locations that occur on branches, or the expectation thereof, have been used in several studies (Lemey et al.; 2014; Bahl et al.; 2011) .

For discrete phylogeography, as a subset of discrete trait analysis (DTA) (Baele et al.; 2018) which can also be applied to other traits (such as in (Faria et al.; 2013)), covariates can now be incorporated (Lemey et al.; 2014). An alternative to DTA is the structured coalescent (De Maio et al.; 2015; Müller et al.; 2017), which may not be sensitive to sampling bias in the same way as DTA (Müller et al.; 2017).

Experimental objectives

Phylogenetic methods were used for ancestral state reconstruction in order to characterize importation events. Specifically, objectives were the following:

1. Apply maximum parsimony ancestral state reconstruction techniques to characterize groups of taxa associated with independent importations of influenza A/H3N2 during the 2018-2019 season, with particular focus on a cluster of samples identified as part of an outbreak investigation.
2. Application of these methods, as well as Bayesian modelling, on large scale SARS-CoV-2 datasets to investigate patterns of importation during the first and second waves of the COVID-19 pandemic.

7.3 Methodology

7.3.1 Datasets

For the influenza analysis, previously described H3N2 datasets were used (see (Southgate et al.; 2020)). For the SARS-CoV-2 analysis, COG-UK (The COVID-19 Genomics UK (COG-UK) consortium; 2020) phylopipe data was used up to 2020-11-09, including 104 568 genome sequences. Sequences obtained from lighthouse labs (LHL; Pillar 2) were excluded due to skewed sampling frequencies over time.

7.3.2 Influenza analysis

In order to characterize potentially imported lineages, ancestral state reconstruction was performed. A phylogeny with Welsh samples and all globally available genomes with 90% coverage of HA and 60% coverage of other segments was constructed with IQTree (Nguyen et al.; 2015). Sequences were downloaded from the NCBI influenza virus resource (Bao et al.; 2008b), comprising 1482

(A/H3N2, 18/19) samples. A GTR substitution model and 2000 ultrafast bootstrap replicates (Hoang et al.; 2018) was used. Multiple sequence alignment was performed as previously. Then, using character states denoting Wales and global states, maximum parsimony ancestral state reconstruction was performed using Sankoff's algorithm (Sankoff; 1975). In order to visualize these assignments, reconstructions were used to decorate the nodes of a small tree constructed of Welsh samples, again using IQTREE (Nguyen et al.; 2015). H3N2 trees were rooted using A/Perth/16/2009 as an outgroup.

7.3.3 SARS-CoV-2 importation analysis

For the first, fast analysis, a full phylogeny was obtained from the COG-UK dataset (The COVID-19 Genomics UK (COG-UK) consortium; 2020), and used gotree (`https://github.com/evolbioinfo/gotree`) to perform ACCTRAN (Swofford and Maddison; 1987) ancestral state reconstruction. Maximal 'imported' subtrees were then extracted at 'Wales' nodes with 'not Wales' direct ancestors, calculated using BioPython (Talevich et al.; 2012). The earliest sample date within a subtree was used as the first realization of a sample from that subtree (not the root date).

For the second analysis, stratified sampling was performed across time (month) and location (Wales, not Wales) to obtain 10,000 sample taxa (max sample date 2020-11-03) from the COG-UK dataset (The COVID-19 Genomics UK (COG-UK) consortium; 2020). A tree was then obtained (also computed by the COG-UK group (The COVID-19 Genomics UK (COG-UK) consortium; 2020)) for these taxa, and used treetime (Sagulenko et al.; 2018) with a fixed clock rate (1.0×10^{-3}) to obtain a time scaled phylogeny. A 2-rate epoch model was then used (Membrebe et al.; 2019), implemented in BEAST v1.10.4 (Suchard et al.; 2018) in order to assess differences in subsitution rate for discrete location trait over time in two broad regimes, high and low restriction epochs, corresponding to stringent lockdown and relaxation. Approximately (to the day), the transition points were 2020-03-29 and 2020-07-17, with rates r_1, r_2, r_1.

7.4 Results

7.4.1 Visualizing influenza importation events

Figure 7.1 gives the H3N2 phylogenetic tree (as in the previous chapter), except with additional ancestral state reconstruction for internal nodes. This analysis indicated that the samples fall into over 30 separately imported groups, many of which were singletons, implying that influenza was seeded into Wales several dozen times in the 2018-2019 season, as opposed to having been fewer, larger outbreaks. This is somewhat expected, given the strong travel links between Wales and England. More importantly, the Usk outbreak samples all fell within a single imported lineage. Without ancestral state reconstruction, as demonstrated in previous chapters, some samples may appear to be epidemiologically related when they are not. Consider the Usk outbreak group (A_Usk_3851, A_Usk_3844, A_Usk_3848, and A_Usk_3845). In figure 7.1, 3 other samples fall within the same imported sub-clade. Firstly, without this resolution, one could only compare these samples with ones from other sub-clades, which for epidemiological investigation, would not be meaningful. Furthermore, one may be tempted to analyze these samples on a level that includes other, neighboring samples. However, under the assumption that the reconstruction is correct, it is possible to rule out, for example A_Newport_1149, as being epidemiologically related to the Usk outbreak.

7.4.2 COVID-19 importation

During the COVID-19 pandemic, over 10,000 SARS-CoV-2 genomes were sequenced by Public Health Wales (the sample distribution of these is given in Figure 7.2). Ancestral state reconstruction was then applied to a larger scale, in order to assess signatures of changes in the rate of importation. As with the influenza dataset, inimported sub-lineages were inferred using global sequences. Figure 7.4 shows the distribution of samples over time, with the distribution of the first observed time of each imported sub-lineage, as well as their ratio. As expected, this ratio was highest during the beginning of the pandemic, and began to decrease until around July. This decrease was expected, since behavioral and policy factors were in place to reduce travel and transmission of the virus. During July and August, which coincided with relaxation of many lock-down measures, a rapid increase in this proportion was observed. Unsurprisingly, the second wave of COVID-19 also followed quickly after. It should be noted that this signature could also be caused by other features of the virus transmission dynamics (see *Discussion*).

In order to further assess an aspect of this data, imported lineages with exactly 2 samples were collected. For three time intervals (before 2020-04-01, before 2020-07-15, and before 2020-11-01), the number of pairs that were concordant for local authority were counted (that is, the pairs that were sampled from the same local authority). As shown in Table 7.1, during the second interval, the proportion that were concordant was maximal. Overall, the three categories were significantly different (Chi-square, $p < 0.0001$). Under some assumptions (see *Discussion*), it is expected that pairs of imported samples chosen should have the same geographical co-localization. However, it should be noted that several other factors could explain this observation.

In order to produce a more robust estimation, and understand the uncertainty associated with the ancestral state reconstruction method, Bayesian estimation was performed for a fixed time-tree (10 000 taxon) analysis with a 2-epoch discrete migration model between Wales and not Wales, specifying 2 rates in 3 epochs (r_1, r_2, r_1), as shown in 7.4. Uncertainty due to the ancestral state reconstructions for a fixed tree was relatively small. The mean posterior rate ratio $(\widehat{r_1/r_2}) = 9.13$ was significantly greater than one with estimated posterior probability ($\hat{P} < 0.0005$), as shown in Figure 7.5. The trends observed were largely consistent with the faster ancestral state reconstruction analysis.

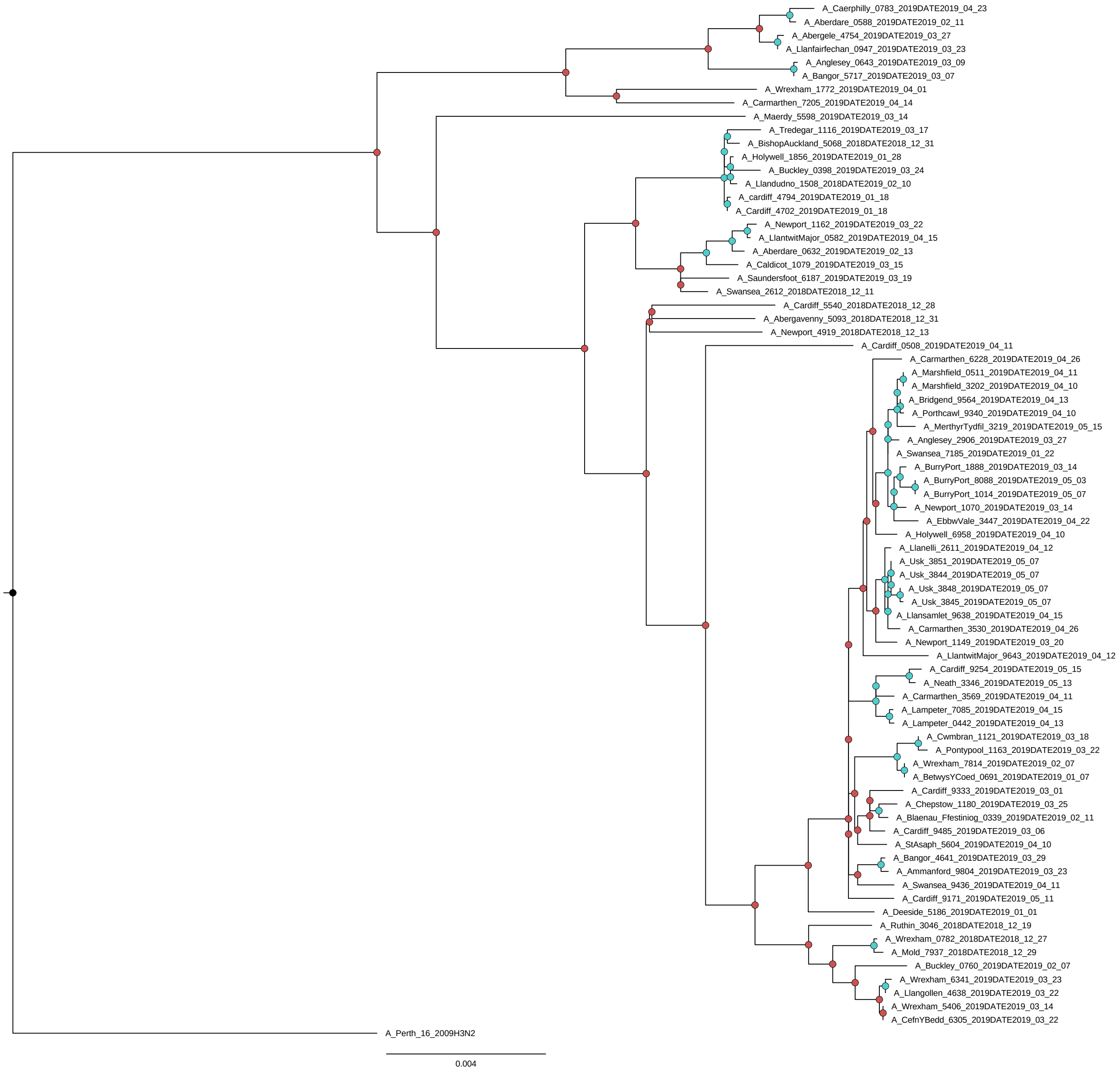

A_Caerphilly_0783_2019DATE2019_04_23
A_Aberdare_0588_2019DATE2019_02_11
A_Abergele_4754_2019DATE2019_03_27
A_Llanfairfechan_0947_2019DATE2019_03_23
A_Anglesey_0643_2019DATE2019_03_09
A_Bangor_5717_2019DATE2019_03_07
A_Wrexham_1772_2019DATE2019_04_01
A_Carmarthen_7205_2019DATE2019_04_14
A_Maerdy_5598_2019DATE2019_03_14
A_Tredegar_1116_2019DATE2019_03_17
A_BishopAuckland_5068_2018DATE2018_12_31
A_Holywell_1856_2019DATE2019_01_28
A_Buckley_0398_2019DATE2019_03_24
A_Llandudno_1508_2018DATE2019_02_10
A_cardiff_4794_2019DATE2019_01_18
A_Cardiff_4702_2019DATE2019_01_18
A_Newport_1162_2019DATE2019_03_22
A_LlantwitMajor_0582_2019DATE2019_04_15
A_Aberdare_0632_2019DATE2019_02_13
A_Caldicot_1079_2019DATE2019_03_15
A_Saundersfoot_6187_2019DATE2019_03_19
A_Swansea_2612_2018DATE2018_12_11
A_Cardiff_5540_2018DATE2018_12_28
A_Abergavenny_5093_2018DATE2018_12_31
A_Newport_4919_2018DATE2018_12_13
A_Cardiff_0508_2019DATE2019_04_11
A_Carmarthen_6228_2019DATE2019_04_26
A_Marshfield_0511_2019DATE2019_04_11
A_Marshfield_3202_2019DATE2019_04_10
A_Bridgend_9564_2019DATE2019_04_13
A_Porthcawl_9340_2019DATE2019_04_10
A_MerthyrTydfil_3219_2019DATE2019_05_15
A_Anglesey_2906_2019DATE2019_03_27
A_Swansea_7185_2019DATE2019_01_22
A_BurryPort_1888_2019DATE2019_03_14
A_BurryPort_8088_2019DATE2019_05_03
A_BurryPort_1014_2019DATE2019_05_07
A_Newport_1070_2019DATE2019_03_14
A_EbbwVale_3447_2019DATE2019_04_22
A_Holywell_6958_2019DATE2019_04_10
A_Llanelli_2611_2019DATE2019_04_12
A_Usk_3851_2019DATE2019_05_07
A_Usk_3844_2019DATE2019_05_07
A_Usk_3848_2019DATE2019_05_07
A_Usk_3845_2019DATE2019_05_07
A_Llansamlet_9638_2019DATE2019_04_15
A_Carmarthen_3530_2019DATE2019_04_26
A_Newport_1149_2019DATE2019_03_20
A_LlantwitMajor_9643_2019DATE2019_04_12
A_Cardiff_9254_2019DATE2019_05_15
A_Neath_3346_2019DATE2019_05_13
A_Carmarthen_3569_2019DATE2019_04_11
A_Lampeter_7085_2019DATE2019_04_15
A_Lampeter_0442_2019DATE2019_04_13
A_Cwmbran_1121_2019DATE2019_03_18
A_Pontypool_1163_2019DATE2019_03_22
A_Wrexham_7814_2019DATE2019_02_07
A_BetwysYCoed_0691_2019DATE2019_01_07
A_Cardiff_9333_2019DATE2019_03_01
A_Chepstow_1180_2019DATE2019_03_25
A_Blaenau_Ffestiniog_0339_2019DATE2019_02_11
A_Cardiff_9485_2019DATE2019_03_06
A_StAsaph_5604_2019DATE2019_04_10
A_Bangor_4641_2019DATE2019_03_29
A_Ammanford_9804_2019DATE2019_03_23
A_Swansea_9436_2019DATE2019_04_11
A_Cardiff_9171_2019DATE2019_05_11
A_Deeside_5186_2019DATE2019_01_01
A_Ruthin_3046_2018DATE2018_12_19
A_Wrexham_0782_2018DATE2018_12_27
A_Mold_7937_2018DATE2018_12_29
A_Buckley_0760_2019DATE2019_02_07
A_Wrexham_6341_2019DATE2019_03_23
A_Llangollen_4638_2019DATE2019_03_22
A_Wrexham_5406_2019DATE2019_03_14
A_CefnYBedd_6305_2019DATE2019_03_22
A_Perth_16_2009H3N2
0.004

Figure 7.1: **Phylogenetic tree for influenza A/H3N2 (2018-2019) with ancestral state reconstructions at internal nodes.** Blue nodes indicate common ancestors inferred to be in Wales, and red those that are elsewhere globally. This reconstruction implies that influenza was seeded into Wales at least several dozen times during the 2018-2019 season. Furthermore, the reconstruction demonstrates a natural way to group samples in terms of potential epidemiological relatedness.

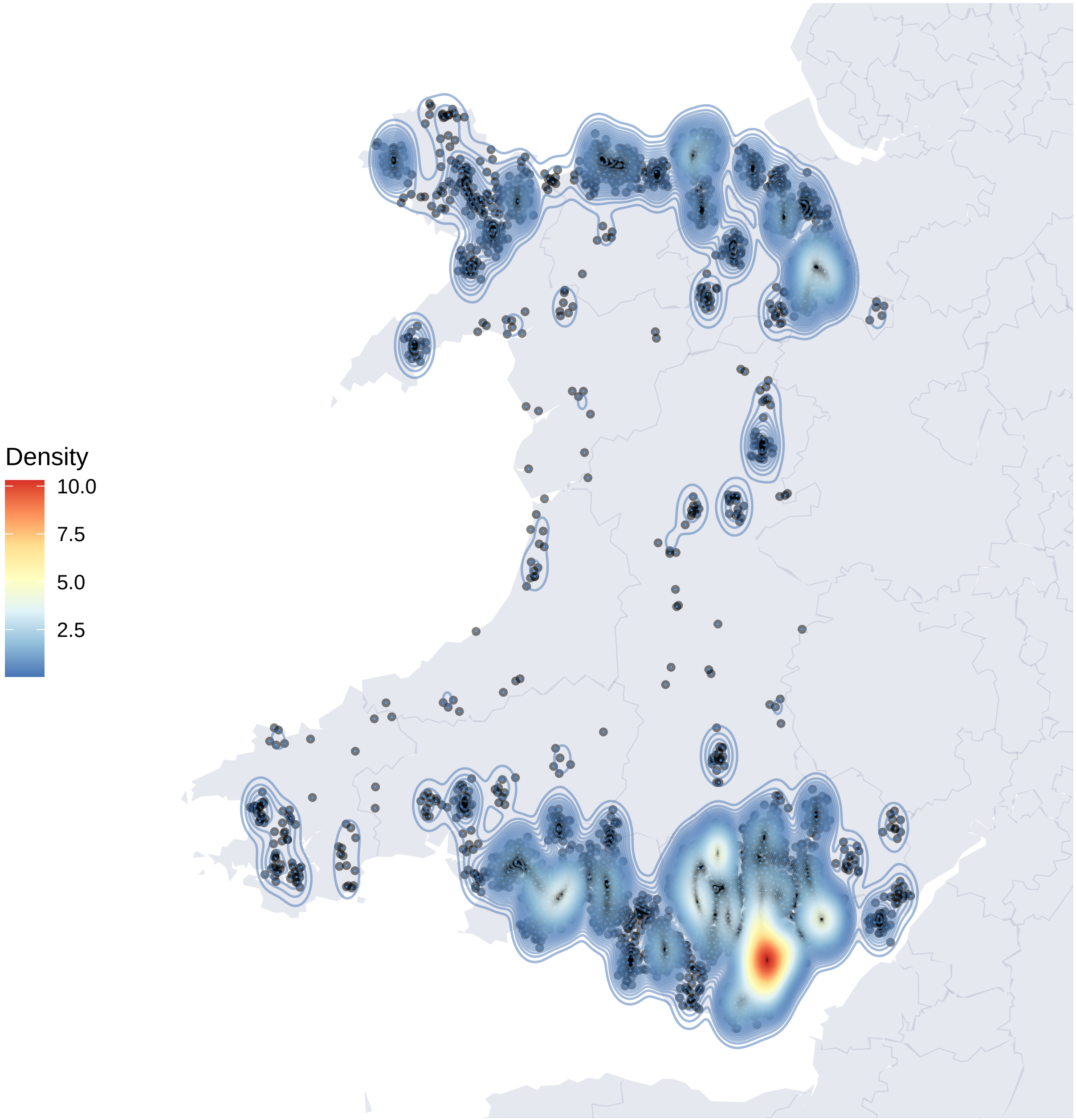


Figure 7.2: **Spatial sampling distribution.** Points were jittered with Gaussian noise. Kernel density estimate derived from unjittered points. The majority of samples came from the South Wales area, in particular around Cardiff. Reflecting population density, most samples were collected either from North or South Wales, with few between.

7.5 Discussion

7.5.1 Importation of influenza A/H3N2 during the 2018-2019 season

Results indicated that it is possible to further identify candidate epidemiological relevance on the basis of sub-lineage identification. It was possible to identify the epidemiologically investigated Usk outbreak as associated with an importation event with 3 other samples, implying that they may have

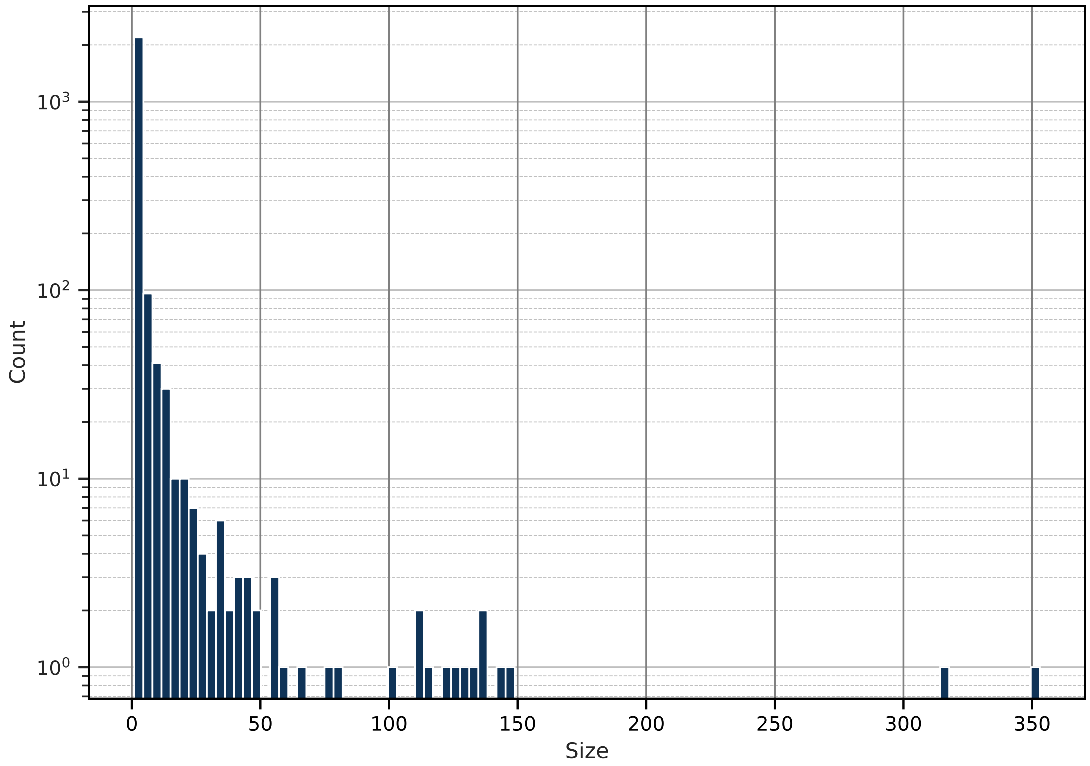


Figure 7.3: **Sub-lineage size distribution**. The vast majority of sub-lineages had a single or very few samples. In comparison, relatively few had hundreds of samples, although two outliers existed with over 300 samples.

been, at least for some time-frame, epidemiologically relevant. This method could also be used to discount epidemiological relationships between samples that appear to be closely related. I also note that, in terms of population structure, influenza A (H3N2) during the 2018/2019 season represented dozens of separate importation events. Given that many of these sub-lineages represented more than one sample (in some cases with quite large), it is expected that a reasonable number of these importation events were captured; if the sampling proportion of imported lineages were poor, one would expect most imported subtrees to be singletons.

7.5.2 Importation dynamics of SARS-CoV-2 during the first and second wave

A similar approach was applied to the first analysis, on a larger scale, to capture the statistical properties of importation over time, and were able to capture broad trends. As expected, importation was highest during the very start of the epidemic when sampling began (also a result of low sampling proportion, and a small database of contextual sequences to allow the ASR procedure to work). Interestingly, signatures of rising importation proportion appeared before the onset of the second wave,

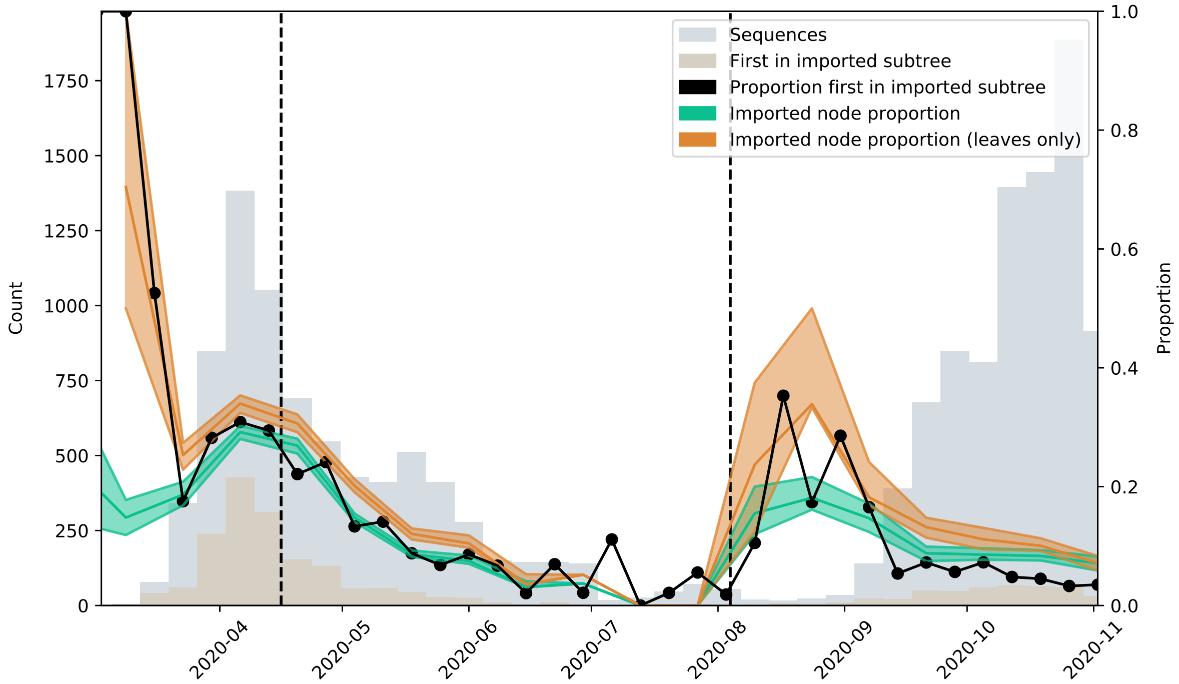


Figure 7.4: **Proportion of lineages imported over time as estimated by ancestral state reconstruction.** Here, the black line gives the number of samples that are the first in some imported sub-lineages out of the total number of sequences in that interval. This is expected to partly reflect the proportion of cases that were imported. As can be seen, during the start of the epidemic, this quotient was highest, and then tapered off, as expected. During July and August, this ratio again began to rise, which coincided with easing of lockdown restrictions. In green, indthe same inference but made with BEAST and the time-scaled phylogeny is indicated, with intervals representing the 95% HPD interval. These intervals were relatively small, indicating that the reconstruction itself represented a relatively small source of uncertainty. In orange is the same BEAST analysis, except only counting leaf transitions. Finally, in the background, grey denotes the sequence counts, and brown the number of samples first in an imported subtree, as by the ASR analysis.

indicating that this analysis could be used as part of routine monitoring during pandemics to help predict a rise in cases. These results indicate that independently imported sub-trees can be identified with reasonable certainty by maximum parsimony, especially for those nodes with high bootstrap support, where there is not expected to be uncertainty in the tree itself. The trends indicated by the maximum parsimony method were largely consistent with those observed from the Bayesian analysis. Furthermore, for 10,000 taxa, the 95% HPD intervals were narrow.

Despite these results, some caveats should be stated. Firstly, utilizing a fixed clock rate of 1×10^{-3} is not likely to be correct, given that substitution rates can vary between populations and time scales. Furthermore, fixed branch lengths were also utilized; a full BEAST analysis usually integrates over branch lengths and topologies. As such, this could mask a considerable amount of uncertainty. However, it should be stated that, at least in the MP ASR analysis, most of the imported subtrees were,

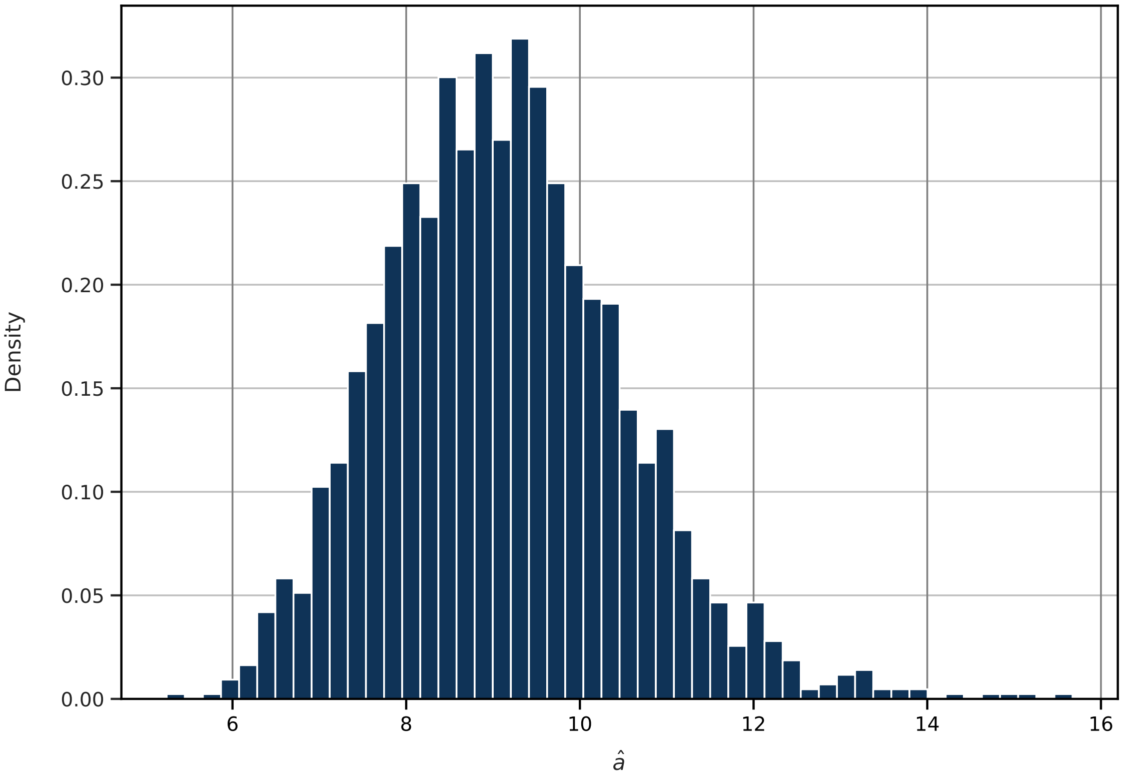


Figure 7.5: **Samples from posterior distribution for rate quotient r_1/r_2.**

Table 7.1: **Contingency table for imported pairs**

Concordant LA	**2020-04-01**	**2020-07-15**	**2020-11-01**
False	32	43	8
True	29	157	18

in fact, singletons, or few taxa. As such, it is expected that many transition events to take place close to or on tips; tips have fixed dates, so these events are invariant to clock rates and branch lengths. It should also be stated that, although we count transitions at nodes, in fact the transitions occur on branches. Methods have been developed to estimate the position along a branch length when a transition occurs (Minin and Suchard; 2008a). The 2-rate epoch model was a coarse simplification of what is expected in reality; that is, that the importation rate changes continuously through time. One could add more rates to improve the resolution of this inference, or in principle use an approach similar to the Bayesian skyride (Minin et al.; 2008). However, I suggest that the 3-epoch model still provided useful support for the changing rate of importation over time reflected in the node proportion statistics. Furthermore, it is expected that this model provides a better fit than a single-rate model. Recently developed methods could be used to assess model fit in this case (Baele et al.; 2012).

7.5.3 Conclusions

The results of this study indicate that ancestral state reconstruction can be used to identify independently imported subtrees during an epidemic, which can assist in epidemiological investigations. Furthermore, I provide evidence for broad trends in importation rate throughout the COVID-19 pandemic, which coincided with the start of the second wave.

7.6 Acknowledgments

I would like to specifically acknowledge the hard work of the large number of individuals involved in the SARS-CoV-2 sequencing initiatives at both Public Health Wales (PHW) and COG-UK. Without the tireless work done by these individuals to produce the datasets used in this study, as well as providing advice and discussion, I would not be have been able to perform this analysis.

Chapter 8

Discussion

8.1 Bioinformatics methods for influenza virus whole genome sequencing pipelines

In this work I have developed bioinformatics strategies for RNA virus whole genome sequencing initiatives. For large-scale projects, hundreds or thousands of sequences can be processed per month, which must be automated. Existing tools can be used to this end, though often they have not been designed for the specific datasets at hand. Often, verifying outputs of existing software can be costly in terms of person hours. Furthermore, often as a new virus is sequenced or the input data is adjusted, such as due to a change in laboratory protocols, or even sequencing instruments, existing software may prove unsuitable. As such, providing optimized software for not just sequence assembly, but also aspects of quality control, or downstream steps, can be valuable, reduce waiting times, and costs associated with manual curation of outputs, as well as offering safeguards by automatic flagging of samples. Furthermore, downstream tasks may also be complex, and receive less attention; defining outbreaks, finding nearest neighbors, and classifying sequences. Often, these tasks are routine for analyses, but can be labor intensive. Here, I developed three software components: `VAPOR`, for reference selection or classification of unassembled reads; `CODETECTEM`, for automated detection of mixed infections or sample contaminations; `NBRFIND`, for range and nearest neighbor queries.

Firstly, I designed `VAPOR`, a program for classifying influenza virus short read data prior to assembly. `VAPOR` was designed to make use of a DBG mapping strategy, where references are queries against a DBG built from raw reads. This process allowed us to retrieve influenza reads from nasal swab samples with carryover from the human host or bacteria, and identify a closely related sequence from a large database with high accuracy. For real data, most of the classifications retrieved a neighbor with $> 99.8\%$ identity to subsequently assembled contigs. Furthermore, I found that when using VAPOR in this way, I had an increased retrieval in the number of reads by up to 13.3%. I believe that this approach allows for most efficient use of the available data, minimizing read loss. In principle, further analysis, such as of variants, may also be affected by loss of reads when choosing a too-distant reference for alignment.

Although VAPOR was applied to HIV, with 79, 448 HIV env sequence obtained from the LANL database (Kuiken et al.; 2001) (data not shown), I believe future work could be used to demonstrate applicabililty to a wider range of viruses, such as SARS-CoV-2. However, the algorithm used was relatively memory intensive; for short influenza segments this was not problematic, and could be run with a few GB in a few minutes on a personal laptop, but for larger genomes, such as SARS-CoV-2, this may require more memory. Furthermore, future work could be used to improve on the algorithm, since much research has recently been performed into data structures for pan-genomicss.

Occasionally, samples can be contaminated, or be isolated from legitimate coinfections. It is therefore important to identify these samples, both for quality control purposes, but also in order to detect clinically or epidemiologically relevant samples. Since, for influenza, mixed infections are the mechanism by which pandemic strains arise, it is important to identify these samples, especially when they have mixed host origin. Furthermore, mixed samples could bias downstream analyses, and lead to chimeric or ambiguous assemblies. As such, I developed a mixture modelling approach using EM for mixed sample identification. In simulation, even with simulated quasispecies, I found that estimates of the mixture parameter π could be made generally to within 1% of the true value, although this did not account for uneven sequencing depth across the genome. Additionally, I identified 10 potential mixtures in the real datasets, 4 of which were found to have a high proportion. I developed a hypothesis test approach to evaluate the probabililty under the null of a given mixture, which was found to perform reasonably well, but was probably too sensitive. I made the additional suggestion that the mixture proportion π should also be larger than some pre-determined threshold, and that candidate coinfections could also be subject to SNV calling with LoFreq (Wilm et al.; 2012).

Future work on CODETECTEM should include implementation in C++ for speed, as well as software functionality for automatically creating test datasets with simulated mixtures of real data, in order to assess good threshold for flagging samples. Furthermore, some mixtures could, in principle, represent more than two genomes; as such, future implementations could leverage approaches from general mixture modelling studies targetting intrahost populations, such as in (Zagordi et al.; 2011).

For nearest neighbor search and range query, I developed an approach based on the diagonal transition algorithm for calculating SNP distances under a cost function where indels are twice as costly as substitutions. I found that, although the cost function was superior to the edit distance (the former was in agreement with SNP distances calculated from a MAFFT (Katoh et al.; 2005) MSA), and speed was on average comparable to that of edlib (Šošić and Šikić; 2017), for some outliers, this approach was considerably slower. Since I required high specificity and sensitivity, I found that basic k-mer filtering was unlikely to be adequate. Furthermore, an approach based on radix trees that I developed was also unlikely to be of much use since the compression achieved was only around 2X, which was not worth the cost associated with tree building and loss of simple parallelizability. On average, this approach was able to search 5000 reference sequences for a single query in under 50 seconds. As such, I expect searching hundreds of thousands of sequences to take around 15 minutes, which may or may not be acceptable given the available CPU resources.

Future work for `NBRFIND` could make use of recent developments in data structures for pan-genomics, as frequently applied in bacterial genomics. However, since ultra-high accuracy is desired, and the ability to account for variable coverage, other methods may also be difficult. One option is to explore approaches such as those used by MUMmer (Marçais et al.; 2018).

To my knowledge, this pipeline is the first where individual software components have been designed specifically for the task. In future, I aim to comlete implementation a pipeline using NextFlow (Di Tommaso et al.; 2017), with each software component acting as a module.

8.2 Whole genome sequencing for virus phylogenetics

With growing datasets and increasingly sophisticated phylogenetic and population genetic methods for analysis of virus genome sequence data, expertise and requirements increase in tandem. Here, I examined aspects of this: data requirements and resolution for phylogeny construction and molecular dating, in order to assess potential and limitations of data and time regimes for routine molecular epidemiology; exponential growth modelling for pandemic viruses; methods for assessing signatures of importation for both examination of imported subtrees for the micro-scale, as well as examination of macro-scale signatures of importation during the first and second wave of the COVID-19 pandemic.

In this section, I firstly examined the resolution achievable for influenza virus phylogenetics with traditional HA-NA sequencing compared to WGS. Whilst it may seem obvious that the whole genome provides more information than 2 genes, without benchmarking the quantitative difference is not known. At the time of writing, the benefits of WGS were initially described, but WGS had not become any way as near as widespread as HA-NA sequencing. I found that, as expected, that more sequences provided better resolution, up to around 5 of the 8. Practically speaking, since WGS often results in dropped segments, this is a reasonable expectation. At this point, bootstrap support for trees became greatest. With HA-NA only, resolution is not sufficient to, for example, resolve outbreak-associated clades. I reiterate the calls for WGS to be performed during routine surveillance of influenza.

Furthermore, with simulated idealized epidemics, typically a few months in duration, I examine the time frames and error in molecular clock estimates obtained using WGS data for influenza. Importantly, I found that the error in estimated TMRCA tended to be on the order of a third of the epidemic duration. As such, I recommend that caution be applied to methods that rely on molecular clock estimates for applications that require precision. For example, in cluster investigation, the TMRCA estimated for a small number of samples should be assumed to be imprecise. Again, this may be as expected, given the considerable uncertainty and assumptions (that are rarely satisfied) that go into molecular clock estimates, however, it is important to quantify. Clock rates estimated from these simulations varied with different tools; error ranged from around approximately 30% for treedater, 3% with treetime, less than 1% with BEAST, and 0.08% for a fixed known tree with BEAST. As such, I found that BEAST performed best for these simulated epidemics, although believe for real data these error rates will be much higher. In addition, I found that with HA and NA alone, posterior distributions sampled by BEAST had much higher variance than those for the whole genome. Whilst these simulations were unrealistic, they represent the best case; performance on data from real eidemics is likely to be worse. Here, in summary, I believe I have made compelling argument for the superiority of WGS over HA-NA sequencing alone.

Next, I performed exponential growth modelling for the first wave of the COVID-19 pandemic using 42 SARS-CoV-2 genomes from the CF postcode area with BEAST. I found a consistent doubling time (approximately a week) with previously reported figures for exponential growth phase. I applied methodology for relating the effective population size $N_e(t)$ to the prevalance $I(t)$. However, I identify several flaws with this procedure: firstly, 95% HPD intervals are extremely wide (801 to 200,000), and secondly, they are sensitive on epidemiological point estimates of the overdispersion parameter k of the offspring distribution. I demonstrate that for smaller values of k, $I(t)$ becomes very large, and variance increases. I believe that these methods can contribute to other methods for estimating exponential growth rates and prevalence using phylodynamics, but should probably not be considered strong evidence. I also note that successful phylodynamic estimation depends on accurate clock rate estimation.

In the final results chapter, I demonstrate the successful application of phylodynamic methods for characterizing importation in two contexts. In the first case, I show that basic methods for ancestral state reconstruction (ASR) can allow us to isolate subtrees that may have been imported, augmenting existing cluster investigation procedures. This procedure can be used to define imported sublineages, and in some cases, exclude the possibility of epidemiological linkage. Although the maximum parsimony ASR procedure may be a heuristic, and may also be sensitive to sampling, in many ways it is a formalization of what is already done by hand; that is, sequenced sampes will be examined by an expert on a tree in context with global samples, and if enough global sequences sit between two candidates, their linkage may be considered unlikely. In many ways, since this study was performed, a procedure of this type has been (and is used at the time of writing) used for defining SARS-CoV-2 UK lineages (The COVID-19 Genomics UK (COG-UK) consortium; 2020). In the second case, I demonstrate how these methods can be used to cature bulk trends over time. I compared the MP ASR method to a more sophisticated epoch model with BEAST. These results were largely consistent; in both cases, spikes in importation were captured immediately preceding the first and second waves. I believe that these methods could be developed over time and be used routinely, possibly to inform policy makers, and assist in the prediction of epidemic growth.

In total, I argue that although these methods can provide powerful insights, phylodynamic methods represent an additional layer of complexity on top of fundamental procedures such as sampling and clock estimation, which can easily go wrong. If these procedures are not specified correctly, the entire procedure will fail too. Robust benchmarking and refinement of methodology is crucial going forward, especially for automation. I also demonstrate that the cutting edge ecosystem of phylodynamic methods only become unlocked once sufficient genome content is sequenced. At least for influenza and SARS-CoV-2, this requires whole genome sequencing, which I recommend as the standard.

References

Abouelhoda, M. I., Kurtz, S. and Ohlebusch, E. (2004). Replacing suffix trees with enhanced suffix arrays, *Journal of discrete algorithms* **2**(1): 53–86.

Abouelhoda, M. I., Kurtz, S. and Ohlebusch, E. (2006). Enhanced suffix arrays and applications, *Handbook of Computational Molecular Biology, Computer and Information Science Series* pp. 7–28.

Ackelsberg, J., Rakeman, J., Hughes, S., Petersen, J., Mead, P., Schriefer, M., Kingry, L., Hoffmaster, A. and Gee, J. E. (2015). Lack of evidence for plague or anthrax on the new york city subway, *Cell systems* **1**(1): 4–5.

Ackermann, H.-W. (2009). Phage classification and characterization, *Bacteriophages*, Springer, pp. 127–140.

Afshinnekoo, E., Meydan, C., Chowdhury, S., Jaroudi, D., Boyer, C., Bernstein, N., Maritz, J. M., Reeves, D., Gandara, J., Chhangawala, S. et al. (2015). Geospatial resolution of human and bacterial diversity with city-scale metagenomics, *Cell systems* **1**(1): 72–87.

Aggarwal, C. C. (2014). *Data classification: algorithms and applications*, 1 edn, Chapman & Hall/CRC press.

Ahn, S. and Vikalo, H. (2017). aBayesQR: A bayesian method for reconstruction of viral populations characterized by low diversity, *International Conference on Research in Computational Molecular Biology*, Springer, pp. 353–369.

Aho, A. V. and Corasick, M. J. (1975). Efficient string matching: an aid to bibliographic search, *Communications of the ACM* **18**(6): 333–340.

Ajami, N. J., Wong, M. C., Ross, M. C., Lloyd, R. E. and Petrosino, J. F. (2018). Maximal viral information recovery from sequence data using VirMAP, *Nature communications* **9**(1): 1–9.

Alavandi, S. and Poornima, M. (2012). Viral metagenomics: a tool for virus discovery and diversity in aquaculture, *Indian Journal of Virology* **23**(2): 88–98.

Allen, L. J. (2017). A primer on stochastic epidemic models: Formulation, numerical simulation, and analysis, *Infectious Disease Modelling* **2**(2): 128–142.

Almodaresi, F., Pandey, P., Ferdman, M., Johnson, R. and Patro, R. (2019). An efficient, scalable and exact representation of high-dimensional color information enabled via de Bruijn graph search, *International Conference on Research in Computational Molecular Biology*, Springer, pp. 1–18.

Almodaresi, F., Pandey, P. and Patro, R. (2017). Rainbowfish: a succinct colored de bruijn graph representation, *17th International Workshop on Algorithms in Bioinformatics (WABI 2017)*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Almutairy, M. and Torng, E. (2018). Comparing fixed sampling with minimizer sampling when using k-mer indexes to find maximal exact matches, *PLOS ONE* **13**(2): e0189960.

Alser, M., Hassan, H., Kumar, A., Mutlu, O. and Alkan, C. (2019). Shouji: a fast and efficient pre-alignment filter for sequence alignment, *Bioinformatics* **35**(21): 4255–4263.

Alser, M., Hassan, H., Xin, H., Ergin, O., Mutlu, O. and Alkan, C. (2017). GateKeeper: a new hardware architecture for accelerating pre-alignment in DNA short read mapping, *Bioinformatics* **33**(21): 3355–3363.

Altman, N. and Krzywinski, M. (2018). The curse(s) of dimensionality, *Nature Methods* **15**: 399–400.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. and Lipman, D. J. (1990). Basic local alignment search tool, *Journal of molecular biology* **215**(3): 403–410.

Andoni, A. and Krauthgamer, R. (2008). The smoothed complexity of edit distance, *International Colloquium on Automata, Languages, and Programming*, Springer, pp. 357–369.

Andoni, A. and Nosatzki, N. S. (2020). Edit distance in near-linear time: it's a constant factor, *arXiv:2005.07678* .

Antipov, D., Raiko, M., Lapidus, A. and Pevzner, P. A. (2020). Metaviral SPAdes: assembly of viruses from metagenomic data, *Bioinformatics* **36**: 4126–4129.

Antoneli, F., Passos, F. M., Lopes, L. R. and Briones, M. R. (2018). A Kolmogorov-Smirnov test for the molecular clock based on bayesian ensembles of phylogenies, *PloS one* **13**(1).

Archer, J., Rambaut, A., Taillon, B. E., Harrigan, P. R., Lewis, M. and Robertson, D. L. (2010). The evolutionary analysis of emerging low frequency HIV-1 CXCR4 using variants through timean ultra-deep approach, *PLoS computational biology* **6**(12).

Arenas, M. (2015). Trends in substitution models of molecular evolution, *Frontiers in Genetics* **6**: 319.

Aris-Brosou, S. and Yang, Z. (2002). Effects of models of rate evolution on estimation of divergence dates with special reference to the metazoan 18S ribosomal RNA phylogeny, *Systematic Biology* **51**(5): 703–714.

Astrovskaya, I., Tork, B., Mangul, S., Westbrooks, K., Măndoiu, I., Balfe, P. and Zelikovsky, A. (2011). Inferring viral quasispecies spectra from 454 pyrosequencing reads, *BMC bioinformatics* **12**: S1.

Ayres, D. L., Darling, A., Zwickl, D. J., Beerli, P., Holder, M. T., Lewis, P. O., Huelsenbeck, J. P., Ronquist, F., Swofford, D. L., Cummings, M. P. et al. (2012). Beagle: an application programming interface and high-performance computing library for statistical phylogenetics, *Systematic biology* **61**(1): 170–173.

Baaijens, J. A., El Aabidine, A. Z., Rivals, E. and Schönhuth, A. (2017). De novo assembly of viral quasispecies using overlap graphs, *Genome research* **27**(5): 835–848.

Baaijens, J. A., Van der Roest, B., Köster, J., Stougie, L. and Schönhuth, A. (2019). Full-length *de novo* viral quasispecies assembly through variation graph construction, *Bioinformatics* **35**(24): 5086–5094.

Baaijens, J., Van der Roest, B., Koester, J., Stougie, L. and Schoenhuth, A. (2018). Full-length *de novo* viral quasispecies assembly through variation graph construction, *BioRxiv* p. 287177.

Baele, G., Dellicour, S., Suchard, M. A., Lemey, P. and Vrancken, B. (2018). Recent advances in computational phylodynamics, *Current opinion in virology* **31**: 24–32.

Baele, G., Li, W. L. S., Drummond, A. J., Suchard, M. A. and Lemey, P. (2012). Accurate model selection of relaxed molecular clocks in bayesian phylogenetics, *Molecular biology and evolution* **30**(2): 239–243.

Baeza-Yates, R. and Gonnet, G. H. (1992). A new approach to text searching, *Communications of the ACM* **35**(10): 74–82.

Bahl, J., Nelson, M. I., Chan, K. H., Chen, R., Vijaykrishna, D., Halpin, R. A., Stockwell, T. B., Lin, X., Wentworth, D. E., Ghedin, E. et al. (2011). Temporally structured metapopulation dynamics and persistence of influenza A H3N2 virus in humans, *Proceedings of the National Academy of Sciences* **108**(48): 19359–19364.

Baillie, G. J., Galiano, M., Agapow, P.-M., Myers, R., Chiam, R., Gall, A., Palser, A. L., Watson, S. J., Hedge, J., Underwood, A. et al. (2011). Evolutionary dynamics of local pandemic H1N1/2009 influenza virus lineages revealed by whole-genome analysis, *Journal of Virology* **86**(1): 11–18.

Baker, D. N. and Langmead, B. (2019). Dashing: fast and accurate genomic distances with hyperloglog, *Genome Biology* **20**(1): 265.

Ball, F. (1986). A unified approach to the distribution of total size and total area under the trajectory of infectives in epidemic models, *Advances in Applied Probability* **18**(2): 289–310.

Bankevich, A., Nurk, S., Antipov, D., Gurevich, A. A., Dvorkin, M., Kulikov, A. S., Lesin, V. M., Nikolenko, S. I., Pham, S., Prjibelski, A. D. et al. (2012). Spades: a new genome assembly algorithm and its applications to single-cell sequencing, *Journal of computational biology* **19**(5): 455–477.

Bao, Y., Bolotov, P., Dernovoy, D., Kiryutin, B., Zaslavsky, L., Tatusova, T., Ostell, J. and Lipman, D. (2008a). The influenza virus resource at the National Center for Biotechnology Information., *Journal of Virology* **82**: 596–601.

Bao, Y., Bolotov, P., Dernovoy, D., Kiryutin, B., Zaslavsky, L., Tatusova, T., Ostell, J. and Lipman, D. (2008b). The influenza virus resource at the national center for biotechnology information, *Journal of virology* **82**(2): 596–601.

Barbezange, C., Jones, L., Blanc, H., Isakov, O., Celniker, G., Enouf, V., Shomron, N., Vignuzzi, M. and van der Werf, S. (2018). Seasonal genetic drift of human influenza a virus quasispecies revealed by deep sequencing, *Frontiers in microbiology* **9**: 2596.

Barik, S., Das, S. and Vikalo, H. (2018). QSdpR: Viral quasispecies reconstruction via correlation clustering, *Genomics* **110**(6): 375–381.

Bedford, T. (2020). Phylodynamic estimation of incidence and prevalence of novel coronavirus (ncov) infections through time, `http://virological.org/t/phylodynamic-estimation-of-incidence-and-prevalence-of-novel-coronavirus-ncov-infections-through-time/391`.

Bedford, T., Cobey, S., Beerli, P. and Pascual, M. (2010). Global migration dynamics underlie evolution and persistence of human influenza A (H3N2), *PLoS Pathog* **6**(5): e1000918.

Bender, M. A., Farach-Colton, M., Johnson, R., Kraner, R., Kuszmaul, B. C., Medjedovic, D., Montes, P., Shetty, P., Spillane, R. P. and Zadok, E. (2012). Don't thrash: How to cache your hash on flash, *PVLDB* **5**(11): 1627–1637.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching, *Communications of the ACM* **18**(9): 509–517.

Berman, A. and Shapiro, L. G. (1998). Selecting good keys for triangle-inequality-based pruning algorithms, *Proceedings 1998 IEEE International Workshop on Content-Based Access of Image and Video Database*, IEEE, pp. 12–19.

Bhuvaneshwar, K., Song, L., Madhavan, S. and Gusev, Y. (2018). viGEN: An open source pipeline for the detection and quantification of viral rna in human tumors, *Frontiers in microbiology* **9**: 1172.

Biggerstaff, M., Cauchemez, S., Reed, C., Gambhir, M. and Finelli, L. (2014). Estimates of the reproduction number for seasonal, pandemic, and zoonotic influenza: a systematic review of the literature, *BMC infectious diseases* **14**(1): 480.

Bloom, B. H. (1970). Space/time trade-offs in hash coding with allowable errors, *Communications of the ACM* **13**(7): 422–426.

Borgelt, C. and Kruse, R. (2004). Shape and size regularization in expectation maximization and fuzzy clustering, *European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 52–62.

Borges, V., Pinheiro, M., Pechirra, P., Guiomar, R. and Gomes, J. P. (2018). INSaFLU: an automated open web-based bioinformatics suite from-reads for influenza whole-genome-sequencing-based surveillance, *Genome Medicine* **10**: 46.

Boroujeni, M., Ehsani, S., Ghodsi, M., HajiAghayi, M. and Seddighin, S. (2018). Approximating edit distance in truly subquadratic time: Quantum and mapreduce, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, pp. 1170–1189.

Bouckaert, R., Heled, J., Kühnert, D., Vaughan, T., Wu, C.-H., Xie, D., Suchard, M. A., Rambaut, A. and Drummond, A. J. (2014). Beast 2: a software platform for bayesian evolutionary analysis, *PLoS computational biology* **10**(4): e1003537.

Bouvier, N. M. and Palese, P. (2008). The biology of influenza viruses, *Vaccine* **26**: D49–D53.

Boyer, R. S. and Moore, J. S. (1977). A fast string searching algorithm, *Communications of the ACM* **20**(10): 762–772.

Bradley, P., Den Bakker, H. C., Rocha, E. P., McVean, G. and Iqbal, Z. (2019). Ultrafast search of all deposited bacterial and viral genomic data, *Nature biotechnology* **37**(2): 152–159.

Breimer, E. A., Goldberg, M. K. and Lim, D. T. (2003). A learning algorithm for the longest common subsequence problem, *Journal of Experimental Algorithmics (JEA)* **8**: 2–1.

Britton, T., Oxelman, B., Vinnersten, A. and Bremer, K. (2002). Phylogenetic dating with confidence intervals using mean path lengths, *Molecular phylogenetics and evolution* **24**(1): 58–65.

Broder, A. Z. (1997). On the resemblance and containment of documents, *Proceedings. Compression and Complexity of Sequences 1997*, IEEE, pp. 21–29.

Brown, R. P. and Yang, Z. (2011). Rate variation and estimation of divergence times using strict and relaxed clocks, *BMC Evolutionary Biology* **11**(1): 271.

Buchfink, B., Xie, C. and Huson, D. H. (2015). Fast and sensitive protein alignment using DIAMOND, *Nature methods* **12**(1): 59.

Buckingham-Jeffery, E., Isham, V. and House, T. (2018). Gaussian process approximations for fast inference from infectious disease data, *Mathematical biosciences* **301**: 111–120.

Burkhard, W. A. and Keller, R. M. (1973). Some approaches to best-match file searching, *Communications of the ACM* **16**(4): 230–236.

Carrat, F., Vergu, E., Ferguson, N. M., Lemaitre, M., Cauchemez, S., Leach, S. and Valleron, A.-J. (2008). Time lines of infection and disease in human influenza: a review of volunteer challenge studies, *American journal of epidemiology* **167**(7): 775–785.

Cespedes, S. P., Seifert, D., Topolsky, I., Metzner, K. J. and Beerenwinkel, N. (2020). V-pipe: a computational pipeline for assessing viral genetic diversity from high-throughput sequencing data, *bioRxiv* p. 2020.06.09.142919.

Chakraborty, D., Das, D., Goldenberg, E., Koucky, M. and Saks, M. (2018). Approximating edit distance within constant factor in truly sub-quadratic time, *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, IEEE, pp. 979–990.

Chaudhuri, S. and Kaushik, R. (2009). Extending autocompletion to tolerate errors, *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pp. 707–718.

Chen, H., Chen, J. and Kalbfleisch, J. D. (2001). A modified likelihood ratio test for homogeneity in finite mixture models, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63**(1): 19–29.

Chen, J. and Cheng, P. (2000). The limiting distribution of the restricted likelihood ratio statistic for finite mixture models, *Chinese Journal of Applied Probability and Statistics* **2**: 159–167.

Chen, J., Huang, J. and Sun, Y. (2019). TAR-VIR: a pipeline for targeted viral strain reconstruction from metagenomic data, *BMC bioinformatics* **20**(1): 305.

Chen, J., Li, P. et al. (2009). Hypothesis test for normal mixture models: The EM approach, *The Annals of Statistics* **37**(5A): 2523–2542.

Chen, J., Zhao, Y. and Sun, Y. (2018). De novo haplotype reconstruction in viral quasispecies using paired-end read guided path finding, *Bioinformatics* **34**(17): 2927–2935.

Chen, X., Tang, J. and Li, W. (2019). Probabilistic operational reliability of composite power systems considering multiple meteorological factors, *IEEE Transactions on Power Systems* **35**(1): 85–97.

Chikhi, R., Limasset, A., Jackman, S., Simpson, J. T. and Medvedev, P. (2014). On the representation of de Bruijn graphs, *International conference on Research in computational molecular biology*, Springer, pp. 35–55.

Chikhi, R. and Rizk, G. (2013). Space-efficient and exact de bruijn graph representation based on a Bloom filter, *Algorithms for Molecular Biology* **8**(1): 22.

Chor, B., Hendy, M. D., Holland, B. R. and Penny, D. (2000). Multiple maxima of likelihood in phylogenetic trees: an analytic approach, *Molecular Biology and Evolution* **17**(10): 1529–1541.

Coburn, B. J., Wagner, B. G. and Blower, S. (2009). Modeling influenza epidemics and pandemics: insights into the future of swine flu (H1N1), *BMC medicine* **7**(1): 30.

Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B. and Hoon, M. J. (2009). Biopython: freely available Python tools for computational molecular biology and bioinformatics, *Bioinformatics* **25**: 1422–1423.

Combe, M. and Sanjuan, R. (2014). Variation in RNA virus mutation rates across host cells, *PLoS pathogens* **10**(1).

Connor, T. R., Loman, N. J., Thompson, S., Smith, A., Southgate, J., Poplawski, R., Bull, M. J., Richardson, E., Ismail, M., Elwood-Thompson, S., Kitchen, C., Guest, M., Bakke, M., Sheppard,

S. K. and Pallen., M. J. (2016). CLIMB (the cloud infrastructure for microbial bioinformatics): an online resource for the medical microbiology community, *Microbial Genomics* **2**: e000086.

Conway, T. C. and Bromage, A. J. (2011). Succinct data structures for assembling large genomes, *Bioinformatics* **27**(4): 479–486.

Cox, N. J. and Subbarao, K. (2000). Global epidemiology of influenza: past and present, *Annual Review of Medicine* **51**: 407–21.

Crochemore, M., Langiu, A. and Rahman, M. S. (2014). Indexing a sequence for mapping reads with a single mismatch, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **372**: 20130167.

Cutler, D. J. (2000). Estimating divergence times in the presence of an overdispersed molecular clock, *Molecular Biology and Evolution* **17**(11): 1647–1660.

da Silva Candido, D., Claro, I. M., de Jesus, J. G., de Souza, W. M., Moreira, F. R. R., Dellicour, S., Mellan, T. A., du Plessis, L., Pereira, R. H. M., da Silva Sales, F. C. et al. (2020). Evolution and epidemic spread of SARS-CoV-2 in brazil, *medRxiv* .

Daciuk, J., Mihov, S., Watson, B. W. and Watson, R. E. (2000). Incremental construction of minimal acyclic finite-state automata, *Computational linguistics* **26**(1): 3–16.

Danesh, G., Elie, B. and Alizon, S. (2020). Phylogeography with whole genomes., `http://virological.org/t/early-phylodynamics-analysis-of-the-covid-19-epidemics-in-france-using-194-genomes-april-10-2020/467`. Accessed: 18-05-20.

Dasgupta, S. and Schulman, L. (2007). A probabilistic analysis of EM for mixtures of separated, spherical gaussians, *Journal of Machine Learning Research* **8**: 203–226.

De Maio, N., Wu, C.-H., OReilly, K. M. and Wilson, D. (2015). New routes to phylogeography: a bayesian structured coalescent approximation, *PLoS Genet* **11**(8): e1005421.

de Silva, E., Ferguson, N. M. and Fraser, C. (2012). Inferring pandemic growth rates from sequence data, *Journal of The Royal Society Interface* **9**(73): 1797–1808.

de Silva, U. C., Tanaka, H., Nakamura, S., Goto, N. and Yasunaga, T. (2012). A comprehensive analysis of reassortment in influenza A virus, *Biology open* **1**(4): 385–390.

Deatherage, D. E. and Barrick, J. E. (2014). Identification of mutations in laboratory-evolved microbes from next-generation sequencing data using breseq, *Engineering and analyzing multicellular systems*, pp. 165–188.

Debbink, K., McCrone, J. T., Petrie, J. G., Truscon, R., Johnson, E., Mantlo, E. K., Monto, A. S. and Lauring, A. S. (2017). Vaccination has minimal impact on the intrahost diversity of H3N2 influenza viruses, *PLoS pathogens* **13**(1): e1006194.

Delcher, A. L., Kasif, S., Fleischmann, R. D., Peterson, J., White, O. and Salzberg, S. L. (1999). Alignment of whole genomes, *Nucleic acids research* **27**(11): 2369–2376.

Delcher, A. L., Phillippy, A., Carlton, J. and Salzberg, S. L. (2002). Fast algorithms for large-scale genome alignment and comparison, *Nucleic acids research* **30**(11): 2478–2483.

Dellicour, S., Durkin, K., Hong, S. L., Vanmechelen, B., Martí-Carreras, J., Gill, M. S., Meex, C., Bontems, S., André, E., Gilbert, M. et al. (2020). A phylodynamic workflow to rapidly gain insights into the dispersal history and dynamics of sars-cov-2 lineages, *Molecular Biology and Evolution* **msaa284**.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society: Series B (Methodological)* **39**(1): 1–22.

DePristo, M. A., Banks, E., Poplin, R., Garimella, K. V., Maguire, J. R., Hartl, C., Philippakis, A. A., Del Angel, G., Rivas, M. A., Hanna, M. et al. (2011). A framework for variation discovery and genotyping using next-generation DNA sequencing data, *Nature genetics* **43**(5): 491.

Dhanabal, S. and Chandramathi, S. (2011). A review of various k-nearest neighbor query processing techniques, *International Journal of Computer Applications* **31**(7): 14–22.

Di, C.-Z. and Liang, K.-Y. (2011). Likelihood ratio testing for admixture models with application to genetic linkage analysis, *Biometrics* **67**(4): 1249–1259.

Di Tommaso, P., Chatzou, M., Floden, E. W., Barja, P. P., Palumbo, E. and Notredame, C. (2017). Nextflow enables reproducible computational workflows, *Nature biotechnology* **35**(4): 316–319.

Didelot, X., Croucher, N. J., Bentley, S. D., Harris, S. R. and Wilson, D. J. (2018). Bayesian inference of ancestral dates on bacterial phylogenetic trees, *Nucleic acids research* **46**(22): e134–e134.

Dijkstra, E. W. et al. (1959). A note on two problems in connexion with graphs, *Numerische mathematik* **1**(1): 269–271.

Drummond, A. J., Ho, S. Y., Phillips, M. J. and Rambaut, A. (2006). Relaxed phylogenetics and dating with confidence, *PLoS biology* **4**(5): e88.

Drummond, A. J., Nicholls, G. K., Rodrigo, A. G. and Solomon, W. (2002). Estimating mutation parameters, population history and genealogy simultaneously from temporally spaced sequence data, *Genetics* **161**(3): 1307–1320.

Drummond, A. J. and Suchard, M. A. (2010). Bayesian random local clocks, or one rate to rule them all, *BMC biology* **8**(1): 114.

Drummond, A., Pybus, O. G. and Rambaut, A. (2003). Inference of viral evolutionary rates from molecular sequences, *Adv Parasitol* **54**: 331–358.

Drummond, A. and Rodrigo, A. G. (2000). Reconstructing genealogies of serial samples under the assumption of a molecular clock using serial-sample UPGMA, *Molecular Biology and Evolution* **17**(12): 1807–1815.

du Plessis, L., McCrone, J. T., Zarebski, A. E., Hill, V., Ruis, C., Gutierrez, B., Raghwani, J., Ashworth, J., Colquhoun, R., Connor, T. R. et al. (2020). Establishment & lineage dynamics of the SARS-CoV-2 epidemic in the UK, *medRxiv* .

Ducatez, M., Sonnberg, S., Hall, R., Peacey, M., Ralston, J., Webby, R. J. and Huang, Q. (2010). Genotyping assay for the identification of 2009–2010 pandemic and seasonal H1N1 influenza virus reassortants, *Journal of virological methods* **168**(1-2): 78–81.

Duchêne, S., Geoghegan, J. L., Holmes, E. C. and Ho, S. Y. (2016). Estimating evolutionary rates using time-structured data: a general comparison of phylogenetic methods, *Bioinformatics* **32**(22): 3375–3379.

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

Endo, A. (2020). Estimating the overdispersion in covid-19 transmission using outbreak sizes outside china, `https://cmmid.github.io/topics/covid19/overdispersion-from-outbreaksize.html`.

Eriksson, N., Pachter, L., Mitsuya, Y., Rhee, S.-Y., Wang, C., Gharizadeh, B., Ronaghi, M., Shafer, R. W. and Beerenwinkel, N. (2008). Viral population estimation using pyrosequencing, *PLoS computational biology* **4**(5): e1000074.

Fahda, A. and Purwarianti, A. (2017). A statistical and rule-based spelling and grammar checker for indonesian text, *2017 International Conference on Data and Software Engineering (ICoDSE)*, IEEE, pp. 1–6.

Falchi, A., Arena, C., Andreoletti, L., Jacques, J., Leveque, N., Blanchon, T., Lina, B., Turbelin, C., Dorleans, Y., Flahault, A. et al. (2008). Dual infections by influenza A/H3N2 and B viruses and by influenza A/H3N2 and A/H1N1 viruses during winter 2007, Corsica Island, France, *Journal of clinical virology* **41**(2): 148–151.

Fan, X., Yuan, Y., Liu, J. S. et al. (2010). The EM algorithm and the rise of computational biology, *Statistical Science* **25**(4): 476–491.

Faria, N. R., Rambaut, A., Suchard, M. A., Baele, G., Bedford, T., Ward, M. J., Tatem, A. J., Sousa, J. D., Arinaminpathy, N., Pépin, J. et al. (2014). The early spread and epidemic ignition of HIV-1 in human populations, *science* **346**(6205): 56–61.

Faria, N. R., Suchard, M. A., Rambaut, A., Streicker, D. G. and Lemey, P. (2013). Simultaneously reconstructing viral cross-species transmission history and identifying the underlying constraints, *Philosophical Transactions of the Royal Society B: Biological Sciences* **368**(1614): 20120196.

Felsenstein, J. (1981). Evolutionary trees from DNA sequences: A maximum likelihood approach, *Journal of Molecular Evolution* **17**: 368–376.

Felsenstein, J. (1985). Phylogenies and the comparative method, *The American Naturalist* **125**(1): 1–15.

Feng, Z. D. and McCulloch, C. E. (1996). Using bootstrap likelihood ratios in finite mixture models, *Journal of the Royal Statistical Society: Series B (Methodological)* **58**(3): 609–617.

Ferragina, P. and Manzini, G. (2000). Opportunistic data structures with applications, *Proceedings 41st Annual Symposium on Foundations of Computer Science*, IEEE, pp. 390–398.

Flouri, T., Izquierdo-Carrasco, F., Darriba, D., Aberer, A. J., Nguyen, L.-T., Minh, B., Von Haeseler, A. and Stamatakis, A. (2015). The phylogenetic likelihood library, *Systematic biology* **64**(2): 356–362.

Frampton, M. and Houlston, R. (2012). Generation of artificial FASTQ files to evaluate the performance of next-generation sequencing pipelines, *PLoS ONE* **7**: e49110.

Franceschini, G. and Muthukrishnan, S. (2007). In-place suffix sorting, *International Colloquium on Automata, Languages, and Programming*, Springer, pp. 533–545.

Fraser, C., Donnelly, C. A., Cauchemez, S., Hanage, W. P., Van Kerkhove, M. D., Hollingsworth, T. D., Griffin, J., Baggaley, R. F., Jenkins, H. E., Lyons, E. J. et al. (2009). Pandemic potential of a strain of influenza A (H1N1): early findings, *science* **324**(5934): 1557–1561.

Fraser, C. and Li, L. M. (2017). Coalescent models for populations with time-varying population sizes and arbitrary offspring distributions, *bioRxiv* p. 131730.

Frost, S. D. and Volz, E. M. (2010). Viral phylodynamics and the search for an effective number of infections, *Philosophical Transactions of the Royal Society B: Biological Sciences* **365**(1548): 1879–1890.

Fu, Y.-X. (2006). Exact coalescent for the Wright–Fisher model, *Theoretical population biology* **69**(4): 385–394.

Galiano, M., Johnson, B. F., Myers, R., Ellis, J., Daniels, R. and Zambon, M. (2012). Fatal cases of influenza A (H3N2) in children: insights from whole genome sequence analysis, *PloS one* **7**(3): e33166.

Gawrychowski, P. (2012). Faster algorithm for computing the edit distance between SLP-compressed strings, *International Symposium on String Processing and Information Retrieval*, Springer, pp. 229–236.

Geil, A., Farach-Colton, M. and Owens, J. D. (2018). Quotient filters: Approximate membership queries on the GPU, *2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, IEEE, pp. 451–462.

Gerstung, M., Beisel, C., Rechsteiner, M., Wild, P., Schraml, P., Moch, H. and Beerenwinkel, N. (2012). Reliable detection of subclonal single-nucleotide variants in tumour cell populations, *Nature communications* **3**(1): 1–8.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions, *The journal of physical chemistry* **81**(25): 2340–2361.

Gillespie, J. H. (1994). *The causes of molecular evolution*, Vol. 2, Oxford University Press On Demand.

Goldenberg, E., Rubinstein, A. and Saha, B. (2020). Does preprocessing help in fast sequence comparisons?, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 657–670.

Goldstein, E. J., Harvey, W. T., Wilkie, G. S., Shepherd, S. J., MacLean, A. R., Murcia, P. R. and Gunson, R. N. (2017). Integrating patient and whole-genome sequencing data to provide insights into the epidemiology of seasonal influenza A(H3N2) viruses, *Microbial Genomics* **2018**: 4.

Goldstein, E. J., Harvey, W. T., Wilkie, G. S., Shepherd, S. J., MacLean, A. R., Murcia, P. R. and Gunson, R. N. (2018). Integrating patient and whole-genome sequencing data to provide insights into the epidemiology of seasonal influenza A(H3N2) viruses, *Microbial Genomics* **4**: –.

Goldwasser, S. and Holden, D. (2017). The complexity of problems in p given correlated instances, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Gorman, O. T., Bean, W. J., Kawaoka, Y. and Webster, R. G. (1990). Evolution of the nucleoprotein gene of influenza a virus., *Journal of virology* **64**(4): 1487–1497.

Görür, D. and Rasmussen, C. E. (2010). Dirichlet process gaussian mixture models: Choice of the base distribution, *Journal of Computer Science and Technology* **25**(4): 653–664.

Gouda, K. and Rashad, M. (2017). Efficient string edit similarity join algorithm, *Computing and Informatics* **36**(3): 683–704.

Goya, R., Sun, M. G., Morin, R. D., Leung, G., Ha, G., Wiegand, K. C., Senz, J., Crisan, A., Marra, M. A., Hirst, M. et al. (2010). SNVMix: predicting single nucleotide variants from next-generation sequencing of tumors, *Bioinformatics* **26**(6): 730–736.

Grossi, R. and Vitter, J. S. (2005). Compressed suffix arrays and suffix trees with applications to text indexing and string matching, *SIAM Journal on Computing* **35**(2): 378–407.

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

Guindon, S. (2010). Bayesian estimation of divergence times from large sequence alignments, *Molecular Biology and Evolution* **27**(8): 1768–1781.

Guindon, S. and Gascuel, O. (2019). Numerical optimization techniques in maximum likelihood tree inference, *Bioinformatics and Phylogenetics*, Springer, pp. 21–38.

Gusfield, D. (1997). *Algorithms on strings, trees, and sequences : computer science and computational biology*, Cambridge University Press.

Haeupler, B., Rubinstein, A. and Shahrasbi, A. (2019). Near-linear time insertion-deletion codes and (1+ ε)-approximating edit distance via indexing, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pp. 697–708.

Hall, M. D., Woolhouse, M. E. and Rambaut, A. (2016). The effects of sampling strategy on the quality of reconstruction of viral population dynamics using Bayesian skyline family coalescent methods: A simulation study, *Virus evolution* **2**(1).

Hanov, S. (2013). Fast and easy Levenshtein distance using a trie, *November* **30**: 4–30.

Hasegawa, M., Kishino, H. and Yano, T.-a. (1985). Dating of the human-ape splitting by a molecular clock of mitochondrial DNA, *Journal of molecular evolution* **22**(2): 160–174.

Hirschberg, D. S. (1975). A linear space algorithm for computing maximal common subsequences, *Communications of the ACM* **18**(6): 341–343.

Ho, S. Y. and Duchêne, S. (2014). Molecular-clock methods for estimating evolutionary rates and timescales, *Molecular ecology* **23**(24): 5947–5965.

Ho, S. Y., Phillips, M. J., Cooper, A. and Drummond, A. J. (2005). Time dependency of molecular rate estimates and systematic overestimation of recent divergence times, *Molecular biology and evolution* **22**(7): 1561–1568.

Hoang, D. T., Chernomor, O., Von Haeseler, A., Minh, B. Q. and Vinh, L. S. (2018). Ufboot2: improving the ultrafast bootstrap approximation, *Molecular biology and evolution* **35**(2): 518–522.

Hobolth, A. and Stone, E. A. (2009). Simulation from endpoint-conditioned, continuous-time Markov chains on a finite state space, with applications to molecular evolution, *The annals of applied statistics* **3**(3): 1204.

Holley, G. and Peterlongo, P. (2012). Blastgraph: Intensive approximate patternmatching in sequence graphs and de Bruijn graphs., *In: Stringology* pp. 53–63.

Holley, G., Wittler, R. and Stoye, J. (2016). Bloom filter trie: an alignment-free and reference-free data structure for pan-genome storage, *Algorithms for Molecular Biology* **11**(1): 1–9.

Holmes, E. C., Ghedin, E., Miller, N., Taylor, J., Bao, Y., St George, K., Grenfell, B. T., Salzberg, S. L., Fraser, C. M., Lipman, D. J. and Taubenberger, J. K. (2005a). Whole-genome analysis of human influenza A virus reveals multiple persistent lineages and reassortment among recent H3N2 viruses, *PLoS Biology* **3**: e300.

Holmes, E. C., Ghedin, E., Miller, N., Taylor, J., Bao, Y., St George, K., Grenfell, B. T., Salzberg, S. L., Fraser, C. M., Lipman, D. J. et al. (2005b). Whole-genome analysis of human influenza a virus reveals multiple persistent lineages and reassortment among recent H3N2 viruses, *PLoS biology* **3**(9): e300.

Holmes, I. H. (2017). Solving the master equation for indels, *BMC bioinformatics* **18**(1): 255.

Holmes, I., Harris, K. and Quince, C. (2012). Dirichlet multinomial mixtures: generative models for microbial metagenomics, *PloS one* **7**: e30126.

Holub, J. and Melichar, B. (2000). Approximate string matching using factor automata, *Theoretical Computer Science* **249**(2): 305–311.

Hölzer, M. and Marz, M. (2017). Software dedicated to virus sequence analysis bioinformatics goes viral, *Advances in Virus Research*, Vol. 99, pp. 233–257.

Hon, W.-K., Lam, T.-W., Sadakane, K., Sung, W.-K. and Yiu, S.-M. (2007). A space and time efficient algorithm for constructing compressed suffix arrays, *Algorithmica* **48**(1): 23–36.

Hopcroft, J. (1971). An n log n algorithm for minimizing states in a finite automaton, *Theory of machines and computations*, Elsevier, pp. 189–196.

Houghton, R., Ellis, J., Galiano, M., Clark, T. W. and Wyllie, S. (2017). Haemagglutinin and neuraminidase sequencing delineate nosocomial influenza outbreaks with accuracy equivalent to whole genome sequencing, *Journal of Infection* **74**(4): 377–384.

Houlihan, C. F., Frampton, D., Ferns, R. B., Raffle, J., Grant, P., Reidy, M., Hail, L., Thomson, K., Mattes, F., Kozlakidis, Z., Pillay, D., Hayward, A. and Nastouli, E. (2018b). Use of whole-genome sequencing in the investigation of a nosocomial influenza virus outbreak, *Journal of Infectious Diseases* **218**: 1485–1489.

Houlihan, C. F., Frampton, D., Ferns, R. B., Raffle, J., Grant, P., Reidy, M., Hail, L., Thomson, K., Mattes, F., Kozlakidis, Z. et al. (2018a). Use of whole-genome sequencing in the investigation of a nosocomial influenza virus outbreak, *The Journal of infectious diseases* **218**(9): 1485–1489.

Howison, M., Coetzer, M. and Kantor, R. (2019). Measurement error and variant-calling in deep Illumina sequencing of HIV, *Bioinformatics* **35**(12): 2029–2035.

Hu, H. and Lee, D. L. (2005). Range nearest-neighbor query, *IEEE Transactions on knowledge and data engineering* **18**(1): 78–91.

Huelsenbeck, J. P. and Ronquist, F. (2001). MRBAYES: Bayesian inference of phylogenetic trees, *Bioinformatics* **17**(8): 754–755.

Hunt, M., Gall, A., Ong, S. H., Brener, J., Ferns, B., Goulder, P., Nastouli, E., Keane, J. A., Kellam, P. and Otto, T. D. (2015). IVA: accurate *de novo* assembly of RNA virus genomes, *Bioinformatics* **31**(14): 2374–2376.

Hyyrö, H. (2001). Explaining and extending the bit-parallel approximate string matching algorithm of Myers, *Technical report*, Dept. of Computer and Information Sciences, University of Tampere, Tampere, Finland.

Hyyrö, H. (2003). A bit-vector algorithm for computing Levenshtein and Damerau edit distances, *Nord. J. Comput.* **10**(1): 29–39.

Hyyrö, H. and Navarro, G. (2005). Bit-parallel witnesses and their applications to approximate string matching, *Algorithmica* **41**(3): 203–231.

Ibrahim, B., McMahon, D. P., Hufsky, F., Beer, M., Deng, L., Le Mercier, P., Palmarini, M., Thiel, V. and Marz, M. (2018). A new era of virus bioinformatics, *Virus research* **251**: 86–90.

Imai, K., Tamura, K., Tanigaki, T., Takizawa, M., Nakayama, E., Taniguchi, T., Okamoto, M., Nishiyama, Y., Tarumoto, N., Mitsutake, K., Murakami, T., Maesaki, S. and Maeda, T. (2018a). Whole genome sequencing of influenza A and B viruses with the MinION sequencer in the clinical setting: A pilot study, *Front. Microbiol.* **9**: 2748.

Imai, K., Tamura, K., Tanigaki, T., Takizawa, M., Nakayama, E., Taniguchi, T., Okamoto, M., Nishiyama, Y., Tarumoto, N., Mitsutake, K. et al. (2018b). Whole genome sequencing of influenza A and B viruses with the MinION sequencer in the clinical setting: a pilot study, *Frontiers in Microbiology* **9**: 2748.

Inagaki, K., Song, M.-S., Crumpton, J.-C., DeBeauchamp, J., Jeevan, T., Tuomanen, E. I., Webby, R. J. and Hakim, H. (2015). Correlation between the interval of influenza virus infectivity and results of diagnostic assays in a ferret model, *The Journal of infectious diseases* **213**(3): 407–410.

Indyk, P. (2001). Algorithmic applications of low-distortion geometric embeddings, *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, IEEE, pp. 10–33.

Iqbal, Z., Caccamo, M., Turner, I., Flicek, P. and McVean, G. (2012). *De novo* assembly and genotyping of variants using colored de Bruijn graphs, *Nature genetics* **44**(2): 226–232.

Ives, A. R. and Zhu, J. (2006). Statistics for correlated data: phylogenies, space, and time, *Ecological Applications* **16**(1): 20–32.

Izquierdo-Carrasco, F. and Stamatakis, A. (2011). Computing the phylogenetic likelihood function out-of-core, *2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum*, IEEE, pp. 444–451.

Jang, J. and Bae, S.-E. (2018). Comparative co-evolution analysis between the HA and NA genes of influenza A virus, *Virology: research and treatment* **9**: 1178122X18788328.

Jasra, A., Holmes, C. C. and Stephens, D. A. (2005). Markov chain Monte Carlo methods and the label switching problem in bayesian mixture modeling, *Statistical Science* pp. 50–67.

Jenkins, G. M., Rambaut, A., Pybus, O. G. and Holmes, E. C. (2002). Rates of molecular evolution in RNA viruses: a quantitative phylogenetic analysis, *Journal of molecular evolution* **54**(2): 156–165.

Ji, S., Li, G., Li, C. and Feng, J. (2009). Efficient interactive fuzzy keyword search, *Proceedings of the 18th international conference on World wide web*, pp. 371–380.

Ji, X., Zhang, Z., Holbrook, A., Nishimura, A., Baele, G., Rambaut, A., Lemey, P. and Suchard, M. A. (2020). Gradients do grow on trees: a linear-time O(N)-dimensional gradient for statistical phylogenetics, *Molecular Biology and Evolution* .

Ji, Y., Shi, Y., Ding, G. and Li, Y. (2011). A new strategy for better genome assembly from very short reads, *BMC bioinformatics* **12**(1): 493.

Jonges, M., Welkers, M. R. A., Jeeninga, R. E., Meijer, A., Schneeberger, P., Fouchier, R. A. M., de Jong, M. D. and Koopmans, M. (2014). Emergence of the virulence-associated PB2 E627K substitution in a fatal human case of highly pathogenic avian influenza virus A(H7N7) infection as determined by Illumina ultra-deep sequencing, *Virology* **88**: 1694–1702.

Jukes, T. H. and Cantor, C. R. (1969). Evolution of protein molecules, *Mammalian Protein Metabolism* **3**: 132.

Kärkkäinen, J. and Ukkonen, E. (1996). Sparse suffix trees, *International Computing and Combinatorics Conference*, Springer, pp. 219–230.

Karp, R. M. and Rabin, M. O. (1987). Efficient randomized pattern-matching algorithms, *IBM journal of research and development* **31**(2): 249–260.

Katoh, K., Asimenos, G. and Toh, H. (2009). Multiple alignment of DNA sequences with MAFFT, *Bioinformatics for DNA sequence analysis*, Springer, pp. 39–64.

Katoh, K., Kuma, K.-i., Toh, H. and Miyata, T. (2005). Mafft version 5: improvement in accuracy of multiple sequence alignment, *Nucleic acids research* **33**(2): 511–518.

Kendal, A., Lee, D., Parish, H., Raines, D., Noble, G. and Dowdle, W. (1979). Laboratory-based surveillance of influenza virus in the united states during the winter of 1977–1978: Ii. isolation of a mixture of A/Victoria-and A/USSR-like viruses from a single person during an epidemic in Wyoming, USA, January 1978, *American journal of epidemiology* **110**(4): 462–468.

Khan, Z., Bloom, J. S., Kruglyak, L. and Singh, M. (2009). A practical algorithm for finding maximal exact matches in large sequence datasets using sparse suffix arrays, *Bioinformatics* **25**(13): 1609–1616.

Kim, D., Langmead, B. and Salzberg, S. L. (2015). HISAT: a fast spliced aligner with low memory requirements, *Nature Methods* **12**: 357–6.

Kim, K., Tandi, T., Choi, J. W., Moon, J. and Kim, M. (2017). Middle east respiratory syndrome coronavirus (MERS-CoV) outbreak in south korea, 2015: epidemiology, characteristics and public health implications, *Journal of Hospital Infection* **95**(2): 207–213.

Kingman, J. F. (1982). On the genealogy of large populations, *Journal of applied probability* **19**(A): 27–43.

Knuth, D. E., Morris, Jr, J. H. and Pratt, V. R. (1977). Fast pattern matching in strings, *SIAM journal on computing* **6**(2): 323–350.

Kobert, K., Stamatakis, A. and Flouri, T. (2017). Efficient detection of repeating sites to accelerate phylogenetic likelihood calculations, *Systematic biology* **66**(2): 205–217.

Koboldt, D. C., Chen, K., Wylie, T., Larson, D. E., McLellan, M. D., Mardis, E. R., Weinstock, G. M., Wilson, R. K. and Ding, L. (2009). VarScan: variant detection in massively parallel sequencing of individual and pooled samples, *Bioinformatics* **25**(17): 2283–2285.

Koren, S., Walenz, B. P., Berlin, K., Miller, J. R., Bergman, N. H. and Phillippy, A. M. (2017). Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation, *Genome research* **27**(5): 722–736.

Koucký, M. and Saks, M. (2020). Constant factor approximations to edit distance on far input pairs in nearly linear time, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 699–712.

Kuhner, M. K., Yamato, J. and Felsenstein, J. (1998). Maximum likelihood estimation of population growth rates based on the coalescent, *Genetics* **149**(1): 429–434.

Kuiken, C., Foley, B., Hahn, B., Marx, P., McCutchan, F., Mellors, J., Wolinsky, S. and Korber, B. (2001). Hiv sequence compendium 2001, *Theoretical Biology and Biophysics Group, Los Alamos National Laboratory, Los Alamos, NM* .

Kumar, S. and Hedges, S. B. (2016). Advances in time estimation methods for molecular data, *Molecular Biology and Evolution* **33**(4): 863–869.

Kurtz, S., Phillippy, A., Delcher, A. L., Smoot, M., Shumway, M., Antonescu, C. and Salzberg, S. L. (2004). Versatile and open software for comparing large genomes, *Genome biology* **5**(2): R12.

Landau, G. M., Myers, E. W. and Schmidt, J. P. (1998). Incremental string comparison, *SIAM Journal on Computing* **27**(2): 557–582.

Landau, G. M. and Vishkin, U. (1988). Fast string matching with k differences, *Journal of Computer and System Sciences* **37**(1): 63–78.

Langley, C. H. and Fitch, W. M. (1974). An examination of the constancy of the rate of molecular evolution, *Journal of molecular evolution* **3**(3): 161–177.

Langmead, B. and Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2, *Nature methods* **9**: 357–359.

Langmead, B., Trapnell, C., Pop, M. and Salzberg, S. L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome, *Genome biology* **10**(3): R25.

Lau, S.-Y., Joseph, S., Chan, K.-H., Chen, H., Patteril, N. A. G., Elizabeth, S. K., Muhammed, R., Baskar, V., Lau, S. K., Kinne, J. et al. (2016). Complete genome sequence of influenza virus H9N2 associated with a fatal outbreak among chickens in Dubai, *Genome announcements* **4**(4).

Lee, H. K., Lee, C. K., Tang, J. W.-T., Loh, T. P. and Koay, E. S.-C. (2016). Contamination-controlled high-throughput whole genome sequencing for influenza A viruses using the MiSeq sequencer, *Scientific reports* **6**: 33318.

Lee, N., Chan, P. K., Lam, W.-y., Cheuk-chun, C. S. and Hui, D. S. (2010). Co-infection with pandemic H1N1 and seasonal H3N2 influenza viruses, *Annals of internal medicine* **152**(9): 618–619.

Lemey, P., Hong, S., Hill, V., Baele, G., Poletto, C., Colizza, V., O'Toole, A., McCrone, J. T., Andersen, K. G., Worobey, M. et al. (2020). Accommodating individual travel history, global mobility, and unsampled diversity in phylogeography: a SARS-CoV-2 case study., *bioRxiv* .

Lemey, P., Rambaut, A., Bedford, T., Faria, N., Bielejec, F., Baele, G., Russell, C. A., Smith, D. J., Pybus, O. G., Brockmann, D. et al. (2014). Unifying viral genetics and human transportation data to predict the global transmission dynamics of human influenza H3N2, *PloS pathog* **10**(2): e1003932.

Lemey, P., Rambaut, A., Drummond, A. J. and Suchard, M. A. (2009). Bayesian phylogeography finds its roots, *PLoS Comput Biol* **5**(9): e1000520.

Leonard, A. S., McClain, M. T., Smith, G. J. D., Wentworth, D. E., Halpin, R. A., Lin, X., Ransier, A., Stockwell, T. B., Das, S. R., Gilbert, A. S., Lambkin-Williams, R., Ginsburg, G. S., Woods, C. W. and Koelle, K. (2016). Deep sequencing of Influenza A virus from a human challenge study reveals a selective bottleneck and only limited intrahost genetic diversification, *Virology* **90**: 11247–11258.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals, *Soviet physics. Doklady* **10**(8): 707–710.

Lewis, N. S., Verhagen, J. H., Javakhishvili, Z., Russell, C. A., Lexmond, P., Westgeest, K. B., Bestebroer, T. M., Halpin, R. A., Lin, X., Ransier, A. et al. (2015). Influenza A virus evolution and spatio-temporal dynamics in Eurasian wild birds: a phylogenetic and phylogeographical study of whole-genome sequence data, *The Journal of general virology* **96**(Pt 8): 2050.

Li, H. (2013). Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM, *arXiv:1303.3997* .

Li, H. (2016). Minimap and miniasm: fast mapping and *de novo* assembly for noisy long sequences, *Bioinformatics* **32**(14): 2103–2110.

Li, H. (2018). Minimap2: pairwise alignment for nucleotide sequences, *Bioinformatics* **34**(18): 3094–3100.

Li, H. and Durbin, R. (2009d). Fast and accurate short read alignment with burrows–wheeler transform, *bioinformatics* **25**(14): 1754–1760.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G. and Durbin, R. (2009a). The sequence alignment/map format and SAMtools, *Bioinformatics* **25**(16): 2078–2079.

Li, H., Ruan, J. and Durbin, R. (2008). Mapping short DNA sequencing reads and calling variants using mapping quality scores, *Genome research* **18**(11): 1851–1858.

Li, P., Chen, J. and Marriott, P. (2009b). Non-finite fisher information and homogeneity: an EM approach, *Biometrika* **96**(2): 411–426.

Li, Q., Guan, X., Wu, P., Wang, X., Zhou, L., Tong, Y., Ren, R., Leung, K. S., Lau, E. H., Wong, J. Y. et al. (2020). Early transmission dynamics in wuhan, china, of novel coronavirus–infected pneumonia, *New England Journal of Medicine* .

Li, R., Li, Y., Fang, X., Yang, H., Wang, J., Kristiansen, K. and Wang, J. (2009c). SNP detection for massively parallel whole-genome resequencing, *Genome research* **19**(6): 1124–1132.

Li, W. and Godzik, A. (2006). Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences, *Bioinformatics* **22**: 1658–1659.

Li, W.-H., Tanimura, M. and Sharp, P. M. (1988). Rates and dates of divergence between aids virus nucleotide sequences., *Molecular biology and evolution* **5**(4): 313–330.

Li, Y., Schofield, E. and Gönen, M. (2019). A tutorial on Dirichlet process mixture modeling, *Journal of Mathematical Psychology* **91**: 128–144.

Li, Y., Wang, H., Nie, K., Zhang, C., Zhang, Y., Wang, J., Niu, P. and Ma, X. (2016). VIP: an integrated pipeline for metagenomics of virus identification and discovery, *Scientific reports* **6**: 23774.

Li, Z., Li, J. and Huo, H. (2018). Optimal in-place suffix sorting, *International Symposium on String Processing and Information Retrieval*, Springer, pp. 268–284.

Limasset, A., Cazaux, B., Rivals, E. and Peterlongo, P. (2016). Read mapping on *de* Bruijn graphs, *Bioinformatics* **17**: 237.

Lindsay, B. G. (1995). Mixture models: theory, geometry and applications, *NSF-CBMS regional conference series in probability and statistics*, JSTOR, pp. i–163.

Liu, B., Guo, H., Brudno, M. and Wang, Y. (2016). deBGA: read alignment with *de* bruijn graph-based seed and extension, *Bioinformatics* **32**: 32243232.

Liu, W., Li, Z.-D., Tang, F., Wei, M.-T., Tong, Y.-G., Zhang, L., Xin, Z.-T., Ma, M.-J., Zhang, X.-A., Liu, L.-J. et al. (2010). Mixed infections of pandemic H1N1 and seasonal H3N2 viruses in 1 outbreak, *Clinical Infectious Diseases* **50**(10): 1359–1365.

Liu, X. and Shao, Y. (2004). Asymptotics for the likelihood ratio test in a two-component normal mixture model, *Journal of Statistical Planning and Inference* **123**(1): 61–81.

Liu, Y., Zhang, L. Y. and Li, J. (2019). Fast detection of maximal exact matches via fixed sampling of query k-mers and bloom filtering of index k-mers, *Bioinformatics* **35**(22): 4560–4567.

Lloyd-Smith, J. O. (2007). Maximum likelihood estimation of the negative binomial dispersion parameter for highly overdispersed data, with applications to infectious diseases, *PloS one* **2**(2).

Lloyd-Smith, J. O., Schreiber, S. J., Kopp, P. E. and Getz, W. M. (2005). Superspreading and the effect of individual variation on disease emergence, *Nature* **438**(7066): 355–359.

Lu, W., Du, X., Hadjieleftheriou, M. and Ooi, B. C. (2014). Efficiently supporting edit distance based string similarity search using B+-trees, *IEEE Transactions on Knowledge and Data Engineering* **26**(12): 2983–2996.

Lycett, S. (2020). Phylogeography with whole genomes., `http://virological.org/t/phylogeography-with-whole-genomes-24-mar-2020/444`. Accessed: 18-05-20.

Maarala, A. I., Bzhalava, Z., Dillner, J., Heljanko, K. and Bzhalava, D. (2018). ViraPipe: scalable parallel pipeline for viral metagenome analysis from next generation sequencing reads, *Bioinformatics* **34**(6): 928–935.

Macalalad, A. R., Zody, M. C., Charlebois, P., Lennon, N. J., Newman, R. M., Malboeuf, C. M., Ryan, E. M., Boutwell, C. L., Power, K. A., Brackney, D. E. et al. (2012). Highly sensitive and specific detection of rare variants in mixed viral populations from massively parallel sequence data, *PLoS computational biology* **8**(3): e1002417.

Malhotra, R., Wu, M. M. S., Rodrigo, A., Poss, M. and Acharya, R. (2015). Maximum likelihood *de novo* reconstruction of viral populations using paired end sequencing data, *arXiv:1502.04239* .

Malkov, Y. A. and Yashunin, D. A. (2018). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs, *IEEE transactions on pattern analysis and machine intelligence* .

Malkov, Y., Ponomarenko, A., Logvinov, A. and Krylov, V. (2014). Approximate nearest neighbor algorithm based on navigable small world graphs, *Information Systems* **45**: 61–68.

Manber, U. and Myers, G. (1993). Suffix arrays: a new method for on-line string searches, *siam Journal on Computing* **22**(5): 935–948.

Manber, U. et al. (1994). Finding similar files in a large file system, *Usenix winter*, Vol. 94, pp. 1–10.

Manzini, G. (1999). The Burrows-Wheeler transform: theory and practice, *International Symposium on Mathematical Foundations of Computer Science*, Springer, pp. 34–47.

Marçais, G., DeBlasio, D., Pandey, P. and Kingsford, C. (2019). Locality-sensitive hashing for the edit distance, *Bioinformatics* **35**(14): i127–i135.

Marçais, G., Delcher, A. L., Phillippy, A. M., Coston, R., Salzberg, S. L. and Zimin, A. (2018). Mummer4: A fast and versatile genome alignment system, *PLoS computational biology* **14**(1): e1005944.

Marçais, G., Pellow, D., Bork, D., Orenstein, Y., Shamir, R. and Kingsford, C. (2017). Improving the performance of minimizers and winnowing schemes, *Bioinformatics* **33**(14): i110–i117.

Marco-Sola, S., Moure López, J. C., Moreto Planas, M. and Espinosa Morales, A. (2020). Fast gap-affine pairwise alignment using the wavefront algorithm, *Bioinformatics* (btaa777): 1–8.

Marschall, T. (2011). Construction of minimal deterministic finite automata from biological motifs, *Theoretical Computer Science* **412**(8-10): 922–930.

Marschall, T. and Rahmann, S. (2009). Efficient exact motif discovery, *Bioinformatics* **25**(12): i356–i364.

McCauley, S. (2019). Approximate similarity search under edit distance using locality-sensitive hashing, *arXiv:1907.01600* .

McCreight, E. M. (1976). A space-economical suffix tree construction algorithm, *Journal of the ACM (JACM)* **23**(2): 262–272.

McCrone, J. T. and Lauring, A. S. (2016). Measurements of intrahost viral diversity are extremely sensitive to systematic errors in variant calling, *Journal of virology* **90**(15): 6884–6895.

McGinnis, J., Laplante, J., Shudt, M. and George, K. S. (2016a). Next generation sequencing for whole genome analysis and surveillance of influenza A viruses, *Journal of Clinical Virology* **79**: 44–50.

McGinnis, J., Laplante, J., Shudt, M. and George, K. S. (2016b). Next generation sequencing for whole genome analysis and surveillance of influenza a viruses, *Journal of Clinical Virology* **79**: 44–50.

McLachlan, G. J., Lee, S. X. and Rathnayake, S. I. (2019). Finite mixture models, *Annual review of statistics and its application* **6**: 355–378.

McLachlan, G. and Khan, N. (2004). On a resampling approach for tests on the number of clusters with mixture model-based clustering of tissue samples, *Journal of multivariate analysis* **90**(1): 90–105.

Medvedev, P., Georgiou, K., Myers, G. and Brudno, M. (2007). Computability of models for sequence assembly, *International Workshop on Algorithms in Bioinformatics*, Springer, pp. 289–301.

Meinel, D. M., Heinzinger, S., Eberle, U., Ackermann, N., Schönberger, K. and Sing, A. (2018a). Whole genome sequencing identifies influenza A H3N2 transmission and offers superior resolution to classical typing methods, *Infection* **46**: 69–76.

Meinel, D. M., Heinzinger, S., Eberle, U., Ackermann, N., Schönberger, K. and Sing, A. (2018b). Whole genome sequencing identifies influenza A H3N2 transmission and offers superior resolution to classical typing methods, *Infection* **46**(1): 69–76.

Melnykov, V., Maitra, R. et al. (2010). Finite mixture models and model-based clustering, *Statistics Surveys* **4**: 80–116.

Melnykov, V. and Melnykov, I. (2012). Initializing the EM algorithm in gaussian mixture models with an unknown number of components, *Computational Statistics & Data Analysis* **56**(6): 1381–1395.

Melsted, P. and Pritchard, J. K. (2011). Efficient counting of k-mers in dna sequences using a bloom filter, *BMC bioinformatics* **12**: 333.

Membrebe, J. V., Suchard, M. A., Rambaut, A., Baele, G. and Lemey, P. (2019). Bayesian inference of evolutionary histories under time-dependent substitution rates, *Molecular biology and evolution* **36**(8): 1793–1803.

Menzel, P., Ng, K. L. and Krogh, A. (2016). Fast and sensitive taxonomic classification for metagenomics with Kaiju, *Nature communications* **7**(1): 1–9.

Minin, V. N., Bloomquist, E. W. and Suchard, M. A. (2008). Smooth skyride through a rough skyline: Bayesian coalescent-based inference of population dynamics, *Molecular biology and evolution* **25**(7): 1459–1471.

Minin, V. N. and Suchard, M. A. (2008a). Counting labeled transitions in continuous-time Markov models of evolution, *Journal of mathematical biology* **56**(3): 391–412.

Minin, V. N. and Suchard, M. A. (2008b). Fast, accurate and simulation-free stochastic mapping, *Philosophical Transactions of the Royal Society B: Biological Sciences* **363**(1512): 3985–3995.

Mishel, P., Ojala, T., Benner, C., Lakspere, T., Bychkov, D., Jalovaara, P., Kakkola, L., Kallio-Kokko, H., Kantele, A., Kankainen, M. et al. (2015). Comparative analysis of whole-genome sequences of influenza A (H1N1) pdm09 viruses isolated from hospitalized and nonhospitalized patients identifies missense mutations that might be associated with patient hospital admissions in Finland during 2009 to 2014, *Genome Announc.* **3**(4): e00676–15.

Mitankin, P., Mihov, S. and Schulz, K. U. (2011). Deciding word neighborhood with universal neighborhood automata, *Theoretical Computer Science* **412**(22): 2340–2355.

Möller, S., du Plessis, L. and Stadler, T. (2018). Impact of the tree prior on estimating clock rates during epidemic outbreaks, *Proceedings of the National Academy of Sciences* **115**(16): 4200–4205.

Morel, B., Flouri, T. and Stamatakis, A. (2017). A novel heuristic for data distribution in massively parallel phylogenetic inference using site repeats, *2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, IEEE, pp. 81–88.

Morens, D. M., Taubenberger, J. K., Harvey, H. A. and Memoli, M. J. (2010). The 1918 influenza pandemic: Lessons for 2009 and the future, *Critical Care Medicine* **38**: e10–e20.

Morris, J. and Pratt, V. (1970). A linear pattern-matching algorithm, *Technical report, University of California, Berkeley, Computation Center* .

Muggli, M. D., Bowe, A., Noyes, N. R., Morley, P. S., Belk, K. E., Raymond, R., Gagie, T., Puglisi, S. J. and Boucher, C. (2017). Succinct colored de Bruijn graphs, *Bioinformatics* **33**(20): 3181–3187.

Müller, N. F., Rasmussen, D. A. and Stadler, T. (2017). The structured coalescent and its approximations, *Molecular biology and evolution* **34**(11): 2970–2981.

Müller, P. and Quintana, F. A. (2004). Nonparametric Bayesian data analysis, *Statistical science* pp. 95–110.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*, MIT press.

Myers, C. A., Kasper, M. R., Yasuda, C. Y., Savuth, C., Spiro, D. J., Halpin, R., Faix, D. J., Coon, R., Putnam, S. D., Wierzba, T. F. et al. (2011). Dual infection of novel influenza viruses A/H1N1 and A/H3N2 in a cluster of cambodian patients, *The American journal of tropical medicine and hygiene* **85**(5): 961–963.

Myers, E. W. (1986). An O(ND) difference algorithm and its variations, *Algorithmica* **1**(1-4): 251–266.

Myers, E. W. (2005). The fragment assembly string graph, *Bioinformatics* **21**: ii79–ii85.

Myers, E. W., Sutton, G. G., Delcher, A. L., Dew, I. M., Fasulo, D. P., Flanigan, M. J., Kravitz, S. A., Mobarry, C. M., Reinert, K. H., Remington, K. A. et al. (2000). A whole-genome assembly of Drosophila, *Science* **287**(5461): 2196–2204.

Myers, G. (1999). A fast bit-vector algorithm for approximate string matching based on dynamic programming, *Journal of the ACM (JACM)* **46**(3): 395–415.

Nagarajan, N. and Kingsford, C. (2010). Giraf: robust, computational identification of influenza reassortments via graph mining, *Nucleic acids research* **39**(6): e34–e34.

Navarro, G. (2001). A guided tour to approximate string matching, *ACM computing surveys (CSUR)* **33**(1): 31–88.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins, *Journal of molecular biology* **48**(3): 443–453.

Nguyen, L.-T., Schmidt, H. A., Von Haeseler, A. and Minh, B. Q. (2015). Iq-tree: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies, *Molecular biology and evolution* **32**(1): 268–274.

Nicholls, S. M., Poplawski, R., Bull, M. J., Underwood, A., Chapman, M., Abu-Dahab, K., Taylor, B., Jackson, B., Rey, S., Amato, R. et al. (2020). Majora: Continuous integration supporting decentralised sequencing for sars-cov-2 genomic surveillance, *bioRxiv* .

Norouzi, M., Punjani, A. and Fleet, D. J. (2012). Fast search in hamming space with multi-index hashing, *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 3108–3115.

Norris, P. M. and Da Silva, A. M. (2016). Monte Carlo Bayesian inference on a statistical model of sub-gridcolumn moisture variability using high-resolution cloud observations. part 1: Method, *Quarterly Journal of the Royal Meteorological Society* **142**(699): 2505–2527.

Ohlebusch, E., Gog, S. and Kügel, A. (2010). Computing matching statistics and maximal exact matches on compressed full-text indexes, *International Symposium on String Processing and Information Retrieval*, Springer, pp. 347–358.

Ondov, B. D., Treangen, T. J., Melsted, P., Mallonee, A. B., Bergman, N. H., Koren, S. and Phillippy, A. M. (2016). MASH: fast genome and metagenome distance estimation using MinHash, *Genome Biology* **17**: 132.

ONS (2020). Population for postcode districts in england and wales, `https://www.ons.gov.uk/aboutus/transparencyandgovernance/freedomofinformationfoi/populationforpostcodedistrictsinenglandandwales`. Accessed: 18-05-20.

Oong, X. Y., Ng, K. T., Tan, J. L., Chan, K. G., Kamarulzaman, A., Chan, Y. F., Sam, I.-C. and Tee, K. K. (2017). Whole-genome phylogenetic analysis of influenza B/Phuket/3073/2013-like viruses and unique reassortants detected in Malaysia between 2012 and 2014, *PloS one* **12**(1): e0170610.

Orton, R. J., Wright, C. F., Morelli, M. J., King, D. J., Paton, D. J., King, D. P. and Haydon, D. T. (2015). Distinguishing low frequency mutations from RT-PCR and sequence errors in viral deep sequencing data, *BMC Genomics* **16**: 299.

Pandey, P., Almodaresi, F., Bender, M. A., Ferdman, M., Johnson, R. and Patro, R. (2018a). Mantis: A fast, small, and exact large-scale sequence-search index, *Cell systems* **7**(2): 201–207.

Pandey, P., Bender, M. A., Johnson, R. and Patro, R. (2018b). Squeakr: an exact and approximate k-mer counting system, *Bioinformatics* **34**(4): 568–575.

Parker, J., Rambaut, A. and Pybus, O. G. (2008). Correlating viral phenotypes with phylogeny: accounting for phylogenetic uncertainty, *Infection, Genetics and Evolution* **8**(3): 239–246.

Parmigiani, G. and Inoue, L. (2009). *Decision theory: Principles and approaches*, Vol. 812, John Wiley & Sons.

Peacey, M., Hall, R. J., Sonnberg, S., Ducatez, M., Paine, S., Nicol, M., Ralston, J. C., Bandaranayake, D., Hope, V., Webby, R. J. et al. (2010). Pandemic (H1N1) 2009 and seasonal influenza A (H1N1) co-infection, New Zealand, 2009, *Emerging infectious diseases* **16**(10): 1618.

Pepe, M. S. (2003). *The statistical evaluation of medical tests for classification and prediction*, Oxford University Press.

Perez-Garcia, F., Vásquez, V., de Egea, V., Catalán, P., Rodríguez-Sánchez, B. and Bouza, E. (2016). Influenza A and B co-infection: a case–control study and review of the literature, *European Journal of Clinical Microbiology & Infectious Diseases* **35**(6): 941–946.

Petrova, V. N. and Russell, C. A. (2018). The evolution of seasonal influenza viruses, *Nature Reviews Microbiology* **16**: 47–60.

Pevzner, P. A., Tang, H. and Waterman, M. S. (2001). An Eulerian path approach to DNA fragment assembly, *Proceedings of the national academy of sciences* **98**(17): 9748–9753.

Ponomarenko, A., Avrelin, N., Naidan, B. and Boytsov, L. (2014). Comparative analysis of data structures for approximate nearest neighbor search, *Data analytics* pp. 125–130.

Poon, A. F., Gustafson, R., Daly, P., Zerr, L., Demlow, S. E., Wong, J., Woods, C. K., Hogg, R. S., Krajden, M., Moore, D. et al. (2016). Near real-time monitoring of HIV transmission hotspots from routine HIV genotyping: an implementation case study, *The lancet HIV* **3**(5): e231–e238.

Posada-Cespedes, S., Seifert, D. and Beerenwinkel, N. (2017). Recent advances in inferring viral diversity from high-throughput sequencing data, *Virus research* **239**: 17–32.

Posada, D. (2001). The effect of branch length variation on the selection of models of molecular evolution, *Journal of Molecular Evolution* **52**(5): 434–444.

Posada, D. and Crandall, K. A. (2001). Selecting models of nucleotide substitution: an application to human immunodeficiency virus 1 (HIV-1), *Molecular biology and evolution* **18**(6): 897–906.

Powell, D. R., Allison, L. and Dix, T. I. (1999). A versatile divide and conquer technique for optimal string alignment, *Information Processing Letters* **70**(3): 127–139.

Prabhakaran, S., Rey, M., Zagordi, O., Beerenwinkel, N. and Roth, V. (2013). HIV haplotype inference using a propagating dirichlet process mixture model, *IEEE/ACM transactions on computational biology and bioinformatics* **11**(1): 182–191.

Prosperi, M. C., Prosperi, L., Bruselles, A., Abbate, I., Rozera, G., Vincenti, D., Solmone, M. C., Capobianchi, M. R. and Ulivi, G. (2011). Combinatorial analysis and algorithms for quasispecies reconstruction using next-generation sequencing, *BMC bioinformatics* **12**(1): 5.

Prosperi, M. C. and Salemi, M. (2012). Qure: software for viral quasispecies reconstruction from next-generation sequencing data, *Bioinformatics* **28**(1): 132–133.

Psomopoulos, F., Diplaris, S. and Mitkas, P. (2004). A finite state automata based technique for protein classification rules induction, *Proc. Second European Workshop Data Mining and Text Mining in Bioinformatics*.

Pupko, T., Pe, I., Shamir, R. and Graur, D. (2000). A fast algorithm for joint reconstruction of ancestral amino acid sequences, *Molecular biology and evolution* **17**(6): 890–896.

Qin, J., Xiao, C., Hu, S., Zhang, J., Wang, W., Ishikawa, Y., Tsuda, K. and Sadakane, K. (2019). Efficient query autocompletion with edit distance-based error tolerance, *The VLDB Journal* pp. 1–25.

Rambaut, A. (2000). Estimating the rate of molecular evolution: incorporating non-contemporaneous sequences into maximum likelihood phylogenies, *Bioinformatics* **16**(4): 395–399.

Rambaut, A. (2020). Phylogenetic analysis of ncov-2019 genomes, `http://virological.org/t/phylodynamic-analysis-of-ncov-2019-genomes-29-jan-2020/353`. Accessed: 18-05-20.

Rambaut, A., Lam, T. T., Max Carvalho, L. and Pybus, O. G. (2016). Exploring the temporal structure of heterochronous sequences using tempest (formerly path-o-gen), *Virus evolution* **2**(1): vew007.

Rayko, M. and Komissarov, A. (2020). Quality control of low-frequency variants in SARS-CoV-2 genomes, *BioRxiv* .

Ren, L., Du, L., Carin, L. and Dunson, D. B. (2011). Logistic stick-breaking process, *Journal of Machine Learning Research* **12**(1).

Ristad, E. S. and Yianilos, P. N. (1998). Learning string-edit distance, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(5): 522–532.

Rith, S., Chin, S., Sar, B., Phalla, Y., Horm, S. V., Ly, S., Buchy, P., Dussart, P. and Horwood, P. F. (2015). Natural co-infection of influenza A/H3N2 and A/H1N1pdm09 viruses resulting in a reassortant A/H3N2 virus, *Journal of Clinical Virology* **73**: 108–111.

Rizzi, R., Beretta, S., Patterson, M., Pirola, Y., Previtali, M., Della Vedova, G. and Bonizzoni, P. (2019). Overlap graphs and de Bruijn graphs: data structures for *de novo* genome assembly in the big data era, *Quantitative Biology* **7**: 278292.

Robbins, K. E., Lemey, P., Pybus, O. G., Jaffe, H. W., Youngpairoj, A. S., Brown, T. M., Salemi, M., Vandamme, A.-M. and Kalish, M. L. (2003). US human immunodeficiency virus type 1 epidemic: date of origin, population history, and characterization of early strains, *Journal of virology* **77**(11): 6359–6366.

Roberts, M., Hayes, W., Hunt, B. R., Mount, S. M. and Yorke, J. A. (2004). Reducing storage requirements for biological sequence comparison, *Bioinformatics* **20**(18): 3363–3369.

Rose, R., Constantinides, B., Tapinos, A., Robertson, D. L. and Prosperi, M. (2016). Challenges in the analysis of viral metagenomes, *Virus Evolution* **2**(2): vew022.

Routh, A., Chang, M. W., Okulicz, J. F., Johnson, J. E. and Torbett, B. E. (2015). CoVaMa: Co-variation mapper for disequilibrium analysis of mutant loci in viral populations using next-generation sequence data, *Methods* **91**: 40–47.

Roy, S., Hartley, J., Dunn, H., Williams, R., Williams, C. A. and Breuer, J. (2019). Whole-genome sequencing provides data for stratifying infection prevention and control management of nosocomial influenza a, *Clinical Infectious Diseases* **69**(10): 1649–1656.

Rozera, G., Abbate, I., Bruselles, A., Vlassi, C., D'Offizi, G., Narciso, P., Chillemi, G., Prosperi, M., Ippolito, G. and Capobianchi, M. R. (2009). Massively parallel pyrosequencing highlights minority variants in the HIV-1 env quasispecies deriving from lymphomonocyte sub-populations, *Retrovirology* **6**(1): 15.

Rufo, M., Martín, J. and Pérez, C. (2010). New approaches to compute bayes factor in finite mixture models, *Computational statistics & data analysis* **54**(12): 3324–3335.

Rutschmann, F. (2006). Molecular dating of phylogenetic trees: a brief review of current methods that estimate divergence times, *Diversity and Distributions* **12**(1): 35–48.

Rutvisuttinunt, W., Chinnawirotpisan, P., Simasathien, S., Shrestha, S. K., Yoon, I., Klungthong, C. and Fernandez, S. (2013). Simultaneous and complete genome sequencing of influenza A and B with high coverage by Illumina MiSeq platform, *Journal of Virological Methods* **193**: 394–404.

Sagulenko, P., Puller, V. and Neher, R. A. (2018). Treetime: Maximum-likelihood phylodynamic analysis, *Virus evolution* **4**(1): vex042.

Sahinalp, S. C., Tasan, M., Macker, J. and Ozsoyoglu, Z. M. (2003). Distance based indexing for string proximity search, *Proceedings 19th International Conference on Data Engineering (Cat. No. 03CH37405)*, IEEE, pp. 125–136.

Salemi, M., De Oliveira, T., Ciccozzi, M., Rezza, G. and Goodenow, M. M. (2008). High-resolution molecular epidemiology and evolutionary history of HIV-1 subtypes in albania, *PloS one* **3**(1).

Salmela, L. and Rivals, E. (2014). LoRDEC: accurate and efficient long read error correction, *Bioinformatics* **30**: 3506–3514.

Saluja, R., Adiga, D., Ramakrishnan, G., Chaudhuri, P. and Carman, M. (2017). A framework for document specific error detection and corrections in Indic OCR, *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Vol. 4, IEEE, pp. 25–30.

Sanderson, M. J. (2002). Estimating absolute rates of molecular evolution and divergence times: a penalized likelihood approach, *Molecular biology and evolution* **19**(1): 101–109.

Sanjuán, R. (2012). From molecular genetics to phylodynamics: evolutionary relevance of mutation rates across viruses, *PLoS pathogens* **8**(5).

Sankoff, D. (1975). Minimal mutation trees of sequences, *SIAM Journal on Applied Mathematics* **28**(1): 35–42.

Schadt, E. E., Sinsheimer, J. S. and Lange, K. (1998). Computational advances in maximum likelihood methods for molecular phylogeny, *Genome Research* **8**(3): 222–233.

Schenk, J. J. and Hufford, L. (2010). Effects of substitution models on divergence time estimates: simulations and an empirical study of model uncertainty using cornales, *Systematic Botany* **35**(3): 578–592.

Schierup, M. H. and Hein, J. (2000). Recombination and the molecular clock, *Molecular Biology and Evolution* **17**(10): 1578–1579.

Schleimer, S., Wilkerson, D. S. and Aiken, A. (2003). Winnowing: local algorithms for document fingerprinting, *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pp. 76–85.

Schrauwen, E. J., Herfst, S., Chutinimitkul, S., Bestebroer, T. M., Rimmelzwaan, G. F., Osterhaus, A. D., Kuiken, T. and Fouchier, R. A. (2011). Possible increased pathogenicity of pandemic (H1N1) 2009 influenza virus upon reassortment, *Emerging infectious diseases* **17**(2): 200.

Sedlazeck, F. J., Rescheneder, P. and von Haeseler, A. (2013). NextGenMap: fast and accurate read mapping in highly polymorphic genomes, *Bioinformatics* **29**: 2790–2791.

Shao, W., Li, X., Goraya, M., Wang, S. and Chen, J.-L. (2017). Evolution of influenza A virus by mutation and re-assortment, *International journal of molecular sciences* **18**(8): 1650.

Shapiro, B., Rambaut, A. and Drummond, A. J. (2006). Choosing appropriate substitution models for the phylogenetic analysis of protein-coding sequences, *Molecular biology and evolution* **23**(1): 7–9.

Shiino, T., Okabe, N., Yasui, Y., Sunagawa, T., Ujike, M., Obuchi, M., Kishida, N., Xu, H., Takashita, E., Anraku, A. et al. (2010). Molecular evolutionary analysis of the influenza A (H1N1) pdm, may–september, 2009: temporal and spatial spreading profile of the viruses in japan, *PLoS One* **5**(6).

Shoemaker, J. S. and Fitch, W. M. (1989). Evidence from nuclear sequences that invariable sites should be considered when sequence divergence is calculated, *Molecular Biology and Evolution* **6**: 270–289.

Shu, Y. and McCauley, J. (2017). GISAID: Global initiative on sharing all influenza data–from vision to reality, *Eurosurveillance* **22**(13): 30494.

Simmonds, P. and Aiewsakun, P. (2018). Virus classification–where do you draw the line?, *Archives of virology* **163**(8): 2037–2046.

Simon, B., Pichon, M., Valette, M., Burfin, G., Richard, M., Lina, B. and Josset, L. (2019). Whole genome sequencing of A (H3N2) influenza viruses reveals variants associated with severity during the 2016–2017 season, *Viruses* **11**(2): 108.

Simpson, J. T. and Durbin, R. (2010). Efficient construction of an assembly string graph using the FM-index, *Bioinformatics* **26**(12): i367–i373.

Sipser, M. (2012). *Introduction to the Theory of Computation*, 1 edn, Cengage Learning.

Smith, S. A. and OMeara, B. C. (2012). treepl: divergence time estimation using penalized likelihood for large phylogenies, *Bioinformatics* **28**(20): 2689–2690.

Šošić, M. and Šikić, M. (2017). Edlib: a C/C++ library for fast, exact sequence alignment using edit distance, *Bioinformatics* **33**(9): 1394–1395.

Southgate, J. A., Bull, M. J., Brown, C. M., Watkins, J., Corden, S., Southgate, B., Moore, C. and Connor, T. R. (2020). Influenza classification from short reads with VAPOR facilitates robust mapping pipelines and zoonotic strain detection for routine surveillance applications, *Bioinformatics* **36**(6): 1681–1688.

Spielman, D. A. and Teng, S.-H. (2009). Smoothed analysis: an attempt to explain the behavior of algorithms in practice, *Communications of the ACM* **52**(10): 76–84.

Stack, J. C., Welch, J. D., Ferrari, M. J., Shapiro, B. U. and Grenfell, B. T. (2010). Protocols for sampling viral sequences to study epidemic dynamics, *Journal of the Royal Society Interface* **7**(48): 1119–1127.

Stamatakis, A. (2019). A review of approaches for optimizing phylogenetic likelihood calculations, *Bioinformatics and Phylogenetics*, Springer, pp. 1–19.

Streicker, D. G., Lemey, P., Velasco-Villa, A. and Rupprecht, C. E. (2012). Rates of viral evolution are linked to host geography in bat rabies, *PLoS pathogens* **8**(5).

Streicker, D. G., Turmelle, A. S., Vonhof, M. J., Kuzmin, I. V., McCracken, G. F. and Rupprecht, C. E. (2010). Host phylogeny constrains cross-species emergence and establishment of rabies virus in bats, *Science* **329**(5992): 676–679.

Suchard, M. A., Lemey, P., Baele, G., Ayres, D. L., Drummond, A. J. and Rambaut, A. (2018). Bayesian phylogenetic and phylodynamic data integration using BEAST 1.10, *Virus evolution* **4**(1): vey016.

Suess, T., Buchholz, U., Dupke, S., Grunow, R., an der Heiden, M., Heider, A., Biere, B., Schweiger, B., Haas, W., Krause, G. et al. (2010). Shedding and transmission of novel influenza virus A/H1N1 infection in householdsGermany, 2009, *American journal of epidemiology* **171**(11): 1157–1164.

Sütterlin, T., Huber, S., Dickhaus, H. and Grabe, N. (2009). Modeling multi-cellular behavior in epidermal tissue homeostasis via finite state machines in multi-agent systems, *Bioinformatics* **25**(16): 2057–2063.

Sutton, T. D., Clooney, A. G., Ryan, F. J., Ross, R. P. and Hill, C. (2019). Choice of assembly software has a critical impact on virome characterisation, *Microbiome* **7**(1): 12.

Swofford, D. L. and Maddison, W. P. (1987). Reconstructing ancestral character states under wagner parsimony, *Mathematical biosciences* **87**(2): 199–229.

Tafalla, M., Buijssen, M., Geets, R. and Vonk Noordegraaf-Schouten, M. (2016). A comprehensive review of the epidemiology and disease burden of influenza B in 9 European countries, *Human vaccines and immunotherapeutics* **12**: 993–1002.

Tajima, F. (1993). Simple methods for testing the molecular evolutionary clock hypothesis., *Genetics* **135**(2): 599–607.

Talevich, E., Invergo, B. M., Cock, P. J. and Chapman, B. A. (2012). Bio. phylo: a unified toolkit for processing, analyzing and visualizing phylogenetic trees in biopython, *BMC bioinformatics* **13**(1): 209.

Tamura, K., Nei, M. and Kumar, S. (2004). Prospects for inferring very large phylogenies by using the neighbor-joining method, *Proceedings of the National Academy of Sciences* **101**(30): 11030–11035.

Tange, O. (2011). GNU parallel - the command-line power tool, *;login: The USENIX Magazine* **36**: 42–47.
URL: *http://www.gnu.org/s/parallel*

Tarhio, J. and Ukkonen, E. (1993). Approximate boyer–moore string matching, *SIAM Journal on Computing* **22**(2): 243–260.

Taubenberger, J. K. and Kash, J. C. (2010). Influenza virus evolution, host adaptation, and pandemic formation, *Cell Host and Microbe* **7**: 440–451.

Tavarè, S. (1986). Some probabilistic and statistical problems in the analysis of DNA sequences, *Lectures on Mathematics in the Life Sciences* **17**: 57–83.

The COVID-19 Genomics UK (COG-UK) consortium (2020). An integrated national scale sars-cov-2 genomic surveillance network, *The Lancet Microbe* .

Thorne, J. L., Kishino, H. and Painter, I. S. (1998). Estimating the rate of evolution of the rate of molecular evolution., *Molecular biology and evolution* **15**(12): 1647–1657.

To, T.-H., Jung, M., Lycett, S. and Gascuel, O. (2016). Fast dating using least-squares criteria and algorithms, *Systematic biology* **65**(1): 82–97.

Töpfer, A., Marschall, T., Bull, R. A., Luciani, F., Schönhuth, A. and Beerenwinkel, N. (2014). Viral quasispecies assembly via maximal clique enumeration, *PLoS computational biology* **10**(3).

Tsang, T. K., Cowling, B. J., Fang, V. J., Chan, K.-H., Ip, D. K., Leung, G. M., Peiris, J. M. and Cauchemez, S. (2015). Influenza a virus shedding and infectivity in households, *The Journal of infectious diseases* **212**(9): 1420–1428.

Ukkonen, E. (1985a). Algorithms for approximate string matching, *Information and control* **64**(1-3): 100–118.

Ukkonen, E. (1985b). Finding approximate patterns in strings, *Journal of algorithms* **6**(1): 132–137.

Ukkonen, E. (1992). Approximate string-matching with q-grams and maximal matches, *Theoretical computer science* **92**(1): 191–211.

Ukkonen, E. (1993). Approximate string-matching over suffix trees, *Annual Symposium on Combinatorial Pattern Matching*, Springer, pp. 228–242.

Ukkonen, E. (1995). On-line construction of suffix trees, *Algorithmica* **14**(3): 249–260.

Vaughan, G. T., Nadeau, S. A., Scir, J. and Stadler, T. (2020). Phylodynamic analyses of outbreaks in China, Italy, Washington State (USA), and the diamond princess, `http://virological.org/t/phylodynamic-analyses-of-outbreaks-in-china-italy-washington-state-usa-and-the-diamond-princess/439`. Accessed: 18-05-20.

Vaughan, T. G., Leventhal, G. E., Rasmussen, D. A., Drummond, A. J., Welch, D. and Stadler, T. (2019). Estimating epidemic incidence and prevalence from genomic data, *Molecular biology and evolution* **36**(8): 1804–1816.

Vilsker, M., Moosa, Y., Nooij, S., Fonseca, V., Ghysens, Y., Dumon, K., Pauwels, R., Alcantara, L. C., Vanden Eynden, E., Vandamme, A.-M. et al. (2019). Genome detective: an automated system for virus identification from high-throughput sequencing data, *Bioinformatics* **35**(5): 871–873.

Volz, E. and Frost, S. (2017). Scalable relaxed clock phylogenetic dating, *Virus evolution* **3**(2).

Volz, E. M., Koelle, K. and Bedford, T. (2013). Viral phylodynamics, *PLoS computational biology* **9**(3): e1002947.

Volz, E. et al. (2020). Report 5: Phylogenetic analysis of SARS-CoV-2, `https://www.imperial.ac.uk/media/imperial-college/medicine/sph/ide/gida-fellowships/Imperial-College-COVID19-phylogenetics-15-02-2020.pdf`. Accessed: 18-05-20.

Vyverman, M., De Baets, B., Fack, V. and Dawyndt, P. (2012). Prospects and limitations of full-text index structures in genome analysis, *Nucleic acids research* **40**(15): 6993–7015.

Vyverman, M., De Baets, B., Fack, V. and Dawyndt, P. (2013). essaMEM: finding maximal exact matches using enhanced sparse suffix arrays, *Bioinformatics* **29**(6): 802–804.

Waddington, T. (2016). *Dynamic construction of trie-based automata for approximate K-mer matching on heterogeneous CPU-GPU systems*, PhD thesis, University of Missouri–Columbia.

Wagner, R. A. and Fischer, M. J. (1974). The string-to-string correction problem, *Journal of the ACM (JACM)* **21**(1): 168–173.

Wald, A. (1950). Statistical decision functions, *Annals of Mathematical Statistics* **20**.

Wan, Y., Renner, D. W., Albert, I. and Szpara, M. L. (2015). VirAmp: a galaxy-based viral genome assembly pipeline, *Gigascience* **4**(1): s13742–015.

Wandelt, S., Deng, D., Gerdjikov, S., Mishra, S., Mitankin, P., Patil, M., Siragusa, E., Tiskin, A., Wang, W., Wang, J. et al. (2014). State-of-the-art in string similarity search and join, *ACM Sigmod Record* **43**(1): 64–76.

Wang, C., Mitsuya, Y., Gharizadeh, B., Ronaghi, M. and Shafer, R. W. (2007). Characterization of mutation spectra with ultra-deep pyrosequencing: application to HIV-1 drug resistance, *Genome research* **17**(8): 1195–1201.

Wang, J., Yang, X., Wang, B. and Liu, C. (2017). Ls-join: Local similarity join on string collections, *IEEE Transactions on Knowledge and Data Engineering* **29**(9): 1928–1942.

Wang, W., Qin, J., Xiao, C., Lin, X. and Shen, H. T. (2012). Vchunkjoin: An efficient algorithm for edit similarity joins, *IEEE Transactions on Knowledge and Data Engineering* **25**(8): 1916–1929.

Wang, Z., Gu, Q., Ning, Y. and Liu, H. (2015). High dimensional EM algorithm: Statistical optimization and asymptotic normality, *Advances in neural information processing systems*, pp. 2521–2529.

Wei, Z., Wang, W., Hu, P., Lyon, G. J. and Hakonarson, H. (2011). SNVer: a statistical tool for variant calling in analysis of pooled or individual next-generation sequencing data, *Nucleic acids research* **39**(19): e132–e132.

Weiner, P. (1973). Linear pattern matching algorithms, *14th Annual Symposium on Switching and Automata Theory (swat 1973)*, IEEE, pp. 1–11.

Westesson, O., Lunter, G., Paten, B. and Holmes, I. (2011). An alignment-free generalization to indels of felsensteins phylogenetic pruning algorithm, *arXiv* **11034347**.

Westesson, O., Lunter, G., Paten, B. and Holmes, I. (2012). Accurate reconstruction of insertion-deletion histories by statistical phylogenetics, *PLoS One* **7**: e34572.

Wilks, S. S. (1938). The large-sample distribution of the likelihood ratio for testing composite hypotheses, *The annals of mathematical statistics* **9**(1): 60–62.

Wilm, A., Aw, P. P. K., Bertrand, D., Yeo, G. H. T., Ong, S. H., Wong, C. H., Khor, C. C., Petric, R., Hibberd, M. L. and Nagarajan, N. (2012). LoFreq: a sequence-quality aware, ultra-sensitive variant caller for uncovering cell-population heterogeneity from high-throughput sequencing datasets, *Nucleic acids research* **40**(22): 11189–11201.

Wiman, Å., Enkirch, T., Carnahan, A., Böttiger, B., Hagey, T. S., Hagstam, P., Fält, R. and Brytting, M. (2019). Novel influenza A (H1N2) seasonal reassortant identified in a patient sample, Sweden, January 2019, *Eurosurveillance* **24**(9).

Wood, D. E., Lu, J. and Langmead, B. (2019). Improved metagenomic analysis with Kraken 2, *Genome biology* **20**(1): 257.

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

Woodhams, M. (2006). Can deleterious mutations explain the time dependency of molecular rate estimates?, *Molecular Biology and Evolution* **23**(12): 2271–2273.

Woolthuis, R. G., Wallinga, J. and van Boven, M. (2017). Variation in loss of immunity shapes influenza epidemics and the impact of vaccination, *BMC Infectious Diseases* **17**: 632.

Wróbel, B., Torres-Puente, M., Jiménez, N., Bracho, M. A., García-Robles, I., Moya, A. and González-Candelas, F. (2006). Analysis of the overdispersed clock in the short-term evolution of hepatitis C virus: using the E1/E2 gene sequences to infer infection dates in a single source outbreak, *Molecular biology and evolution* **23**(6): 1242–1253.

Wu, J. T., Leung, K. and Leung, G. M. (2020). Nowcasting and forecasting the potential domestic and international spread of the 2019-nCoV outbreak originating in Wuhan, China: a modelling study, *The Lancet* **395**(10225): 689–697.

Wu, N. C., Young, A. P., Al-Mawsawi, L. Q., Olson, C. A., Feng, J., Qi, H., Chen, S.-H., Lu, I.-H., Lin, C.-Y., Chin, R. G., Luan, H. H., Nguyen, N., Nelson, S. F., Li, X., Wu, T.-T. and Sun, R. (2014). High-throughput profiling of influenza A virus hemagglutinin gene at single-nucleotide resolution, *Scientific Reports* **4**: 4942.

Wu, S. and Manber, U. (1992). Agrep–a fast approximate pattern-matching tool, *Usenix Winter 1992 Technical Conference*, pp. 153–162.

Wu, S., Manber, U. and Myers, G. (1996). A subquadratic algorithm for approximate limited expression matching, *Algorithmica* **15**(1): 50–67.

Wüthrich, D., Lang, D., Müller, N. F., Neher, R. A., Stadler, T. and Egli, A. (2019). Evaluation of two workflows for whole genome sequencing-based typing of influenza A viruses, *Journal of virological methods* **266**: 30–33.

Wymant, C., Blanquart, F., Golubchik, T., Gall, A., Bakker, M., Bezemer, D., Croucher, N. J., Hall, M., Hillebregt, M., Ong, S. H., Ratmann, O., Albert, J., Bannert, N., Fellay, J., Fransen, K., Gourlay, A., Grabowski, M. K., Gunsenheimer-Bartmeyer, B., Günthard, H. F., Kivelä, P., Kouyos, R., Laeyendecker, O., Liitsola, K., Meyer, L., Porter, K., Ristola, M., van Sighem, A., Berkhout, B., Cornelissen, M., Kellam, P., Reiss, P., Fraser, C. and BEEHIVE Collaboration (2018). Easy and accurate reconstruction of whole HIV genomes from short-read sequence data with shiver, *Virus Evolution* **4**: vey007.

Xue, K. S., Moncla, L. H., Bedford, T. and Bloom, J. D. (2018). Within-host evolution of human influenza virus, *Trends in microbiology* **26**(9): 781–793.

Yang, J., Meng, X. and Hlavacek, W. S. (2010). Rule-based modelling and simulation of biochemical systems with molecular finite automata, *IET systems biology* **4**(6): 453–466.

Yang, X., Charlebois, P., Gnerre, S., Coole, M. G., Lennon, N. J., Levin, J. Z., Qu, J., Ryan, E. M., Zody, M. C. and Henn, M. R. (2012). De novo assembly of highly diverse viral populations, *BMC genomics* **13**(1): 475.

Yang, X., Charlebois, P., Macalalad, A., Henn, M. R. and Zody, M. C. (2013). V-Phaser 2: variant inference for viral populations, *BMC genomics* **14**(1): 674.

Yang, Z. (1994). Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods, *Journal of Molecular Evolution* **39**: 306–314.

Yi, X. and Caramanis, C. (2015). Regularized EM algorithms: A unified framework and statistical guarantees, *Advances in Neural Information Processing Systems*, pp. 1567–1575.

Yu, G., Smith, D. K., Zhu, H., Guan, Y. and Lam, T. T.-Y. (2017). ggtree: an R package for visualization and annotation of phylogenetic trees with their covariates and other associated data, *Methods in Ecology and Evolution* **8**(1): 28–36.

Yu, X., Jin, T., Cui, Y., Pu, X., Li, J., Xu, J., Liu, G., Jia, H., Liu, D., Song, S., Yu, Y., Xie, L., Huang, R., Ding, H., Kou, Y., Zhou, Y., Wang, Y., Xu, X., Yin, Y., Wang, J., Guo, C., Yang, X., Hu, L., Wu, X., Wang, H., Liu, J., Zhao, G., Zhou, J., Pan, J., Gao, G. F., Yang, R. and Wang, J. (2014).

Influenza H7N9 and H9N2 viruses: Coexistence in poultry linked to human H7N9 infection and genome characteristics, *Virology* **88**: 3423–3431.

Zagordi, O., Bhattacharya, A., Eriksson, N. and Beerenwinkel, N. (2011). ShoRAH: estimating the genetic diversity of a mixed sample from next-generation sequencing data, *BMC bioinformatics* **12**(1): 119.

Zagordi, O., Geyrhofer, L., Roth, V. and Beerenwinkel, N. (2010). Deep sequencing of a genetically heterogeneous sample: local haplotype reconstruction and read error correction, *Journal of computational biology* **17**(3): 417–428.

Zerbino, D. R. and Birney, E. (2008). Velvet: algorithms for *de novo* short read assembly using de Bruijn graphs, *Genome research* **18**(5): 821–829.

Zhang, B. and Srihari, S. N. (2002). A fast algorithm for finding k-nearest neighbors with non-metric dissimilarity, *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*, IEEE, pp. 13–18.

Zhang, H. and Zhang, Q. (2017). Embedjoin: Efficient edit similarity joins via embeddings, *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 585–594.

Zhang, H. and Zhang, Q. (2019). Minjoin: Efficient edit similarity joins via local hash minima, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1093–1103.

Zheng, Y., Gao, S., Padmanabhan, C., Li, R., Galvez, M., Gutierrez, D., Fuentes, S., Ling, K.-S., Kreuze, J. and Fei, Z. (2017). Virusdetect: An automated pipeline for efficient virus discovery using deep sequencing of small RNAs, *Virology* **500**: 130–138.

Zhou, B., Donnelly, M. E., Scholes, D. T., St George, K., Hatta, M., Kawaoka, Y. and Wentworth, D. E. (2009). Single-reaction genomic amplification accelerates sequencing and vaccine production for classical and swine origin human influenza A viruses, *Journal of Virology* **83**: 10309–13.

Zhou, B., Lin, X., Wang, W., Halpin, R. A., Bera, J., Stockwell, T. B., Barr, I. G. and Wentworth, D. E. (2014). Universal influenza B virus genomic amplification facilitates sequencing, diagnostics, and reverse genetics, *Journal of Clinical Microbiology* **52**: 1330–1337.

Zou, X. H., Chen, W. B., Xiang, Z., Zhu, W. F., Lei, Y., Wang, D. Y. and Shu, Y. L. (2016). Evaluation of a single-reaction method for whole genome sequencing of influenza a virus using next generation sequencing, *Biomedical and Environmental Sciences* **29**(1): 41–46.

Zuckerkandl, E. and Pauling, L. (1962). Molecular disease, evolution, and genetic heterogeneity, *Horizons in biochemistry* pp. 189–225.