WILDEBEAST

Riaz Moola

4th Year Project Report
Artificial Intelligence and Computer Science
School of Informatics
University of Edinburgh

2014

Abstract

The 2009 H1N1 pandemic presented new challenges in the evolutionary analysis of rapidly evolving pathogens. Genomes of new strains of the virus were continually shared in real time during the pandemic, and Bayesian phylogenetics tools were manually run to discover, with increasing accuracy, important evolutionary parameters of the pandemic.

This project introduces the WILDEBEAST service which overcomes these challenges to allow optimal real time characterisation of viral pathogens. WILDEBEAST implements a novel set of algorithms that initiate Markov Chain Monte Carlo inferences implemented by the Bayesian phylogentics tool BEAST to carry out this characterisation automatically. WILDEBEAST is deployed as a webservice that allows policy makers to be informed on the virulence and origin of pathogens, and also allows users to manage how the system functions. WILDEBEAST was evaluated on real world data from epidemics caused by H1N1, H3N2, SARS, and dengue, and was able to characterise each of these pathogens in real time simulations.

Acknowledgements

I would like to thank my supervisor, Andrew Rambaut, for providing sound guidance and assistance at every step of the project. I am also grateful that he agreed to supervise me as student from outside his department. I consider the opportunity to study viral pathogens with cutting edge tools like BEAST a once in a lifetime experience, and am excited to have participated in such work while an undergraduate student.

I would also like to thank Ben Murrell and Sasha Murrell for urging me to take this project, and also for helping me understand several bioinformatics concepts.

Finally, I would like to thank my parents for supporting my studies at the University of Edinburgh. I would not have been able to complete a project like this at my previous universities.

Table of Contents

Chapter 1

Introduction

Section 1.1 introduces core concepts in the modern study of epidemics. Section 1.2 then sets forth the motivations for the project, and Section 1.3 presents a summary of the main results.

1.1 Introduction to evolutionary epidemiology

1.1.1 The study of epidemics

The terror evoked by outbreaks of viral pathogens is captured by the writings of Giovanni Boccaccio, who in 1348 wrote "*. . . such terror was struck into the hearts of men and women by this calamity, that brother abandoned brother, and the uncle his nephew, and the sister her brother, and very often the wife her husband.*" [1]. This description was given to events surrounding one of the most potent pandemics in human history - the Black Death - which between 1347 and 1351 is estimated to have claimed the lives of up to half the population of Europe [2].

Our understanding of viral pathogens that cause epidemics and pandemics (an epidemic affecting people over a much larger area, usually globally) has increased immensely since the discovery of the genetic code. Whilst the 14th-century layman may have believed the Black Death to be caused by gods, gypsies, or even earthquakes, modern sequencing technology coupled with probabilistic bioinformatics methods has allowed us to retrospectively analyse genomic data from the dead to determine the true cause of this pandemic - the bacterium *Yersinia pestis* [2]. Today, these sequencing and modelling methods are ubiquitous in the study of epidemics and the agents that cause them.

1.1.2 The age of genomic plenty

The swine-origin influenza A H1N1 (H1N12009) virus emerged in early March 2009 in Mexico and was classified as a pandemic in June 2009 [16]. Studies estimate that, within 12 months of the pandemic, 201 200 deaths had occurred

globally [19]. The rise of the web-based platforms for information dispersal, including rapid, even real time, social networking sites such as Twitter, is reflected in the way epidemics are monitored and reported today. The 2009 pandemic was the first to place evolutionary epidemiologists in an environment where newly sequenced genomes of isolates of the virus were shared in real time through public repositories such as the The Global Initiative on Sharing Avian Influenza Data (GISAID) EpiFlu^{TM}for database, National Center for Biotechnology Information(NCBI) GenBank database [9]. As the pandemic progressed, the cumulative knowledge gleaned through analysis of these genomes, which represent the genetic make-up of different strains of a virus in a number of human hosts, allowed researchers to make inferences about the virus with increasing confidence.

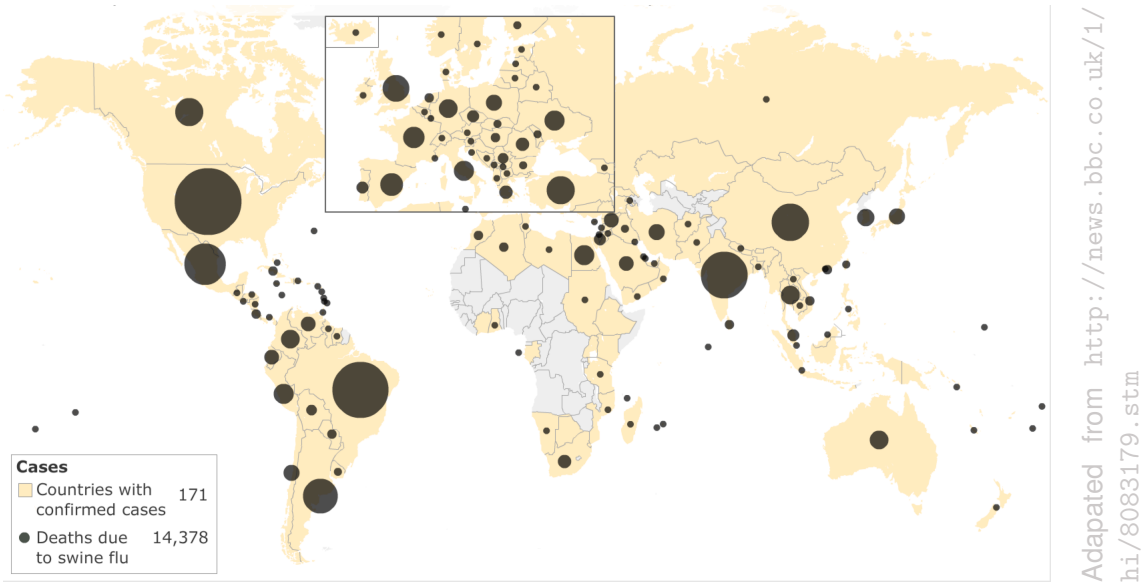


Figure 1.1: H1N12009 Deaths and incidents as confirmed by the European Centre for Disease Prevention and Control up to January 21 2010

1.1.3 Real time characterisation of epidemics

Analysis of the continuous stream of isolate data, using methods and tools from evolutionary epidemiology, made it possible for researchers to infer real time estimates of important evolutionary parameters of the H1N1 virus. This real time characterisation is attractive, as it allows policy makers to be informed rapidly, facilitating swift implementation of policy that defines public response to an epidemic. Appropriate responses through informed policy making are critical in preventing further loss of human life during outbreaks [18].

A number of publicly accessible websites were set up to support rapid dissemination of knowledge gleaned through analysis of incoming data throughout H1N12009. One such website was maintained by the University of Edinburgh Molecular Evolution, Phylogenetics and Epidemiology research group, located at `http://tree.bio.ed.ac.uk/groups/influenza`. A number of epidemiologists and evolutionary biologists collaborated through this site to carry out probabilistic analysis of

sequence data during the epidemic.

The Bayesian Evolutionary Analysis by Sampling Trees (BEAST) software package was a core tool for carrying out these analyses. BEAST is a Bioinformatics software package that allows for probabilistic inference of evolutionary and epidemiological parameters of a virus, through a Bayesian Metropolis Hastings Markov Chain Monte Carlo (MCMC) framework. These inferences are carried out by learning the evolutionary dynamics of a pathogen from sequence data [7]. **Chapter 2** describes the evolutionary model used in BEAST, and how it is used to carry out inference.

1.2 Project rationale

This project is motivated by challenges faced by researches during early attempts at characterisation of the H1N12009 virus. The main goal of the project is to provide a proof of concept for an intelligent webservice that addresses these challenges, allowing it to be realistically deployed during future epidemics to enable optimal, even automatic, early characterisation of viral pathogens. A system dubbed WILDEBEAST - a **W**ebservice for monitoring **I**nfectious **L**ive **D**isease **E**pidemics with **BEAST** was built to meet this goal. The name is inspired by the Afrikaans name for a type of antelope discovered in South Africa - the wildebeest.

1.2.1 Translational Bioinformatics

Translational Bioinformatics (TBI) is an new, broadly defined field, that encompasses methods that seek to extract knowledge from voluminous genomic data and disseminate such knowledge to a wide range of stakeholders, including the public and public health policy makers. [23]. Analysis carried out using BEAST contributed to two major breakthroughs within just three months of detection of H1N1. The first described the origins of the virus as a reassortment of several viruses that has previous been circulating in pigs. The second, a World Health Organisation (WHO) Pandemic Assessment Collaboration report, quantified the infectiousness of the virus, by providing estimates of important evolutionary parameters - for example, the basic reproductive rate (R_0) [16, 17]. R_0, a central concept in epidemiology, is defined as the average number of secondary infections one infection by a viral pathogen causes. Estimates of R_0 directly inform decisions by the WHO and other policy makers about whether to close schools or restrict air travel, for example.

BEAST remains the state of the art tool for evolutionary epidemiology, and is likely to continue playing a major role in future WHO Pandemic Assessment reports. A partially autonomous webservice that allows BEAST to be optimally used in an epidemic environment to provide publicly accessible reports on the best estimates of evolutionary parameters, such as R_0, is in line with the aims of TBI. Such a system provides a web-based interface to allow authenticated researchers to collaboratively analyse sequence data and the results of runs, and can of significant use in the preparation of future pandemic assessment reports.

The web facing front end of WILDEBEAST is introduced introduced in **Chapter 3**, with more details about its architecture presented in **Chapter 4**.

1.2.2 Intelligent Analysis

Genomic sequences representing new cases of infection with a virus are made available (usually by public health staff who upload these sequences to public repositories) during the course of an epidemic at an unpredictable rate. As new sequences become available, additional BEAST MCMC analyses were manually started on the new set of accumulated sequence data. The computational requirements of these analyses are exponential in the number of sequences used, with some runs taking weeks to complete. **Chapter 5** describes and evaluates models that were learnt in order to predict the runtime of BEAST analyses from features of sequence data.

High-throughput sequencing technologies are advancing at such a rate that it is expected that whole genomes of all sampled cases of a pathogen will be made available for analysis during an epidemic [9]. This deluge of data highlights the need for methods that can intelligently downsample this data, to enable inferences within a timescale that can facilitate policy making. Several novel subset selection algorithms are proposed and evaluated in **Chapter 6**.

New data often arrives while older analyses are still proceeding, and no method exists for deciding how to include new data, which may require discarding or halting old analyses. Given the nature of the stochastic inference algorithm in BEAST (MCMC), analysis must be allowed to run for significant periods (up to weeks in real settings) before giving credible parameter estimates, and if analysis are always halted on the arrival of new data, a pathogen cannot be studied in real time. Methods are also needed for deciding which evolutionary estimates to report from a set of ongoing analysis at varying stages of completion, run on disparate sets of sequence data . **Chapter 7** presents a novel sequence insertion procedure and decision framework to address these issues.

WILDEBEAST was evaluated under simulations of previous epidemics, using real sequence data and replicating their release over time. The system was shown to function autonomously, under a range of different modes, and was shown to automatically characteristic the viral pathogens in real time. **Chapter 8** presents these evaluations.

1.3 Results summary

A brief overview of the achievements of the project is presented in this section. Please refer to Chapter 9 for a more detailed discussion of these contributions.

Main results and contributions

- A system for real time characterisation of infectious disease epidemics was implemented - the first of its kind. This system can function autonomously to

analyse sequence data during an epidemic and report estimates of important evolutionary parameters. It is available immediate deployment, and is accessible online at `http://kimura.bio.ed.ac.uk:8080/WILDEBEAST/index`

- The backend of this service implements decision algorithms to overcome challenges of analysing epidemics while they are ongoing. A novel sequence insertion algorithm that allows analysis to converge faster was implemented along with decision procedures. The operation of the system can be fully specified by 9 user-specified settings.
- The BEAST software suite, which implements all evolutionary analyses, was provided. A suite of scripts were implemented to generate and carry out approximately 250 BEAST runs to serve as training data. In total, over 500 runs were analysed during the project, having a total CPU runtime of over 150 days. Novel prediction algorithms were learnt from this data.
- Five algorithms for sequence selection were proposed and implemented, and are shown to overcome challenges in analysing large datasets. A Python pipeline was implemented to carry out a number of experiments to evaluate these methods.
- The entire system was evaluated on real datasets taken from epidemics caused by four viral pathogens - SARS, H1N1, H3N2, and Dengue. The system was shown to characterise all pathogens in real time, even on the largest and most challenging epidemic sequence dataset to date.

Structure of report

A more detailed summary of the layout of the project report is as follows:

- **Chapter 2** presents key background concepts in molecular and evolutionary epidemiology, and presents the details of BEAST, Markov Chain Monte Carlo, and the evolutionary parameters that characterise a viral pathogen.
- **Chapter 3** provides an introduction to the front end of the WILDEBEAST web service, with examples of basic user interactions in defining epidemics for monitoring and managing BEAST analyses, data, and estimates given by the system.
- **Chapter 4** explores the technical architecture and implementation of the web service, and how the system interfaces with BEAST.
- **Chapter 5** describes the generation of training data from which models were learnt for predicting the expected run time and quality of results for BEAST analyses. The results for the basic evaluation of these methods are presented.
- **Chapter 6** discusses the problem of sequence selection, describes four novel sequence selection algorithms and provides the results of several experiments evaluating each methods effectiveness.

- **Chapter 7** describes the decision framework for WILDEBEAST which comprises of a number of automated algorithms for or generating report summaries and methods for starting and stopping BEAST analyses during an epidemic. A sequence-insertion method for reducing the time-lag of parameter estimates is also described and evaluated.
- **Chapter 8** presents evaluations of the entire WILDEBEAST system operating in several different modes and under simulations of several previous epidemics. These evaluations test the system as a whole as well as WILDEBEAST's ability to operate autonomously under real epidemic settings.
- **Chapter 9** outlines the conclusions and directions for future work.

Chapter 2

Background

This chapter first presents the scientific context for the project in Sections 2.1 to 2.6 which provide a detailed introduction to key concepts in Bayesian phylogentics , BEAST, and phylogenetic inference with Markov Chain Monte Carlo. Section 2.6 discusses and critiques previous related work.

2.1 Phylogenetics

The notion of a tree of life is a recurring motif in a number of cultures. Egyptian mythology portrays the emergence of key deities Osiris and Isis from an acacia tree - the representation of life and death, whilst Yggdrasil, the world tree, is a central concept in Norse mythology [3]. A new sense for the term emerged in the 1800s, with the publication of *The Origin of Species*, in which Darwin described the evolutionary relationship of all life as *"the great Tree of life ... with its ever branching and beautiful ramifications"* [4]. The contemporary formalisation of this idea is the phylogenetic tree, now a ubiquitous concept in computational biology.

The use of phylogenies extend beyond simply representing relationships between species on the tree of life. Phylogenies are used to describe histories of populations, genealogical relationships of cells or genes, and even the evolution of language [5]. The most pertinent use of phylogenies to this project is their role in the analysis of emerging viral pathogens, specifically for answering questions about the virulence and origin of such epidemics. Figure 2.1 gives such an example; this phylogeny shows the evolutionary history of simian immunodeficiency viruses (SIVs) and human immunodeficiency virus type 1 and 2. The evolutionary history of the primate lentiviruses reflect a key find in the study of HIV - that HIV-1 and HIV-2 arose from distinct, independent cross-species transmission events [24].

2.2 Molecular sequence data

The advent of next generation sequencing techniques has made it possible to rapidly determine the DNA sequences of isolates during a particular epidemic. The ordered sequence of nucleotides define the complete genomeof an organism.

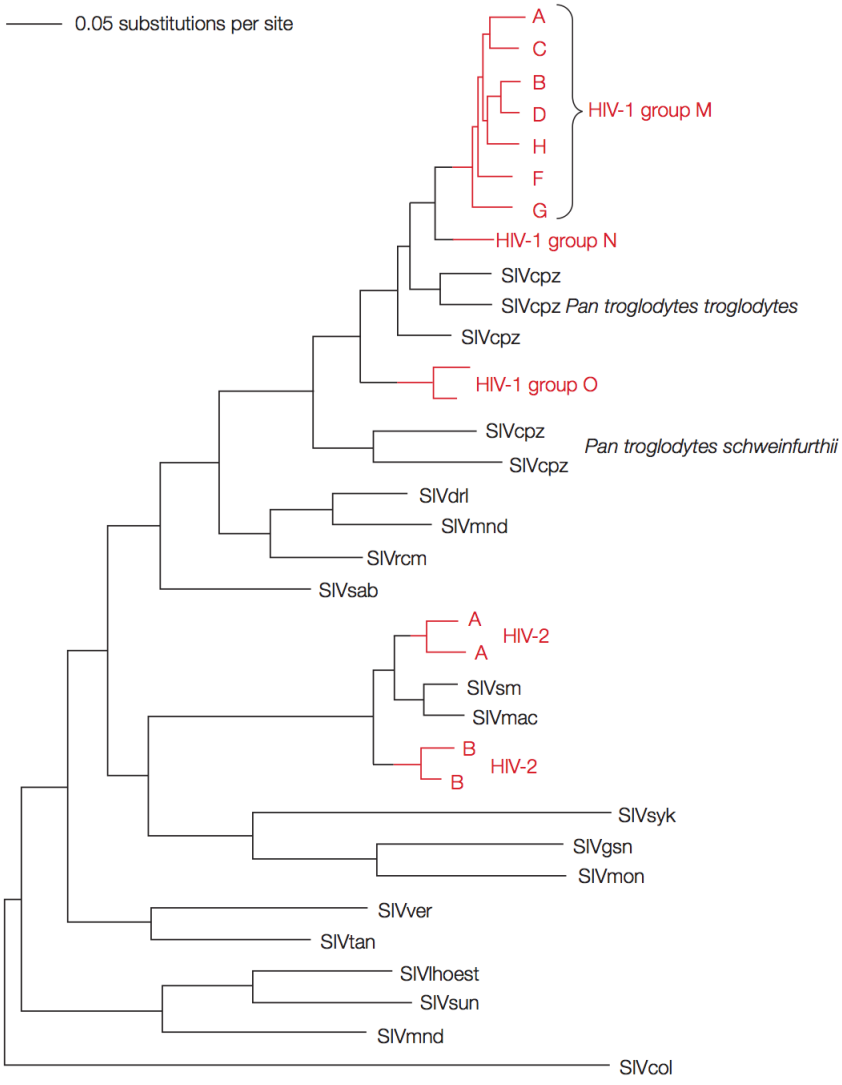


Figure 2.1: Evolutionary history of the primate lentiviruses HIV and SIV.

Each nucleotide takes on the value of one of four bases, adenine (often abbreviated to an "A", and replaced by Uracil or "U" in RNA), thymine ("T"), cytosine ("C"), or guanine ("G"). A complete genome of an isolate of a pathogen will be referred to as a sequence. Given a set of sequences - each of a different strain of the pathogen - multiple sequence alignment is carried out to align the columns of sequences so that each column represents a homologous site and hence all nucleotides at that position in the alignment have a shared ancestry. While sequence alignment is a core problem in bioinformatics, it is not usually an issue with short time-scale virus data.

These sequence alignments serve as input to molecular phylogenetics algorithms, which infer the evolutionary relationship between these sequences and encode this information in a phylogeny and other parameters - - in such a process the phylogeny is not our main interest, but part of the analysis [5]. A number of methods exist for inferring phylogenies, such as parsimony, maximum likelihood, and

distance-based approaches, but Bayesian Markov chain Monte Carlo (MCMC) is considered the state-of-the-art method for the construction of phylogenies from sequence data [6].

2.3 Bayesian Evolutionary Analysis by Sampling Trees

Bayesian Evolutionary Analysis by Sampling Trees (BEAST) is a popular software package that provides a highly configurable Bayesian MCMC framework for phylogenetic inference of evolutionary parameters and hypothesis testing of evolutionary models from molecular sequence data [7]. BEAST has found wide use in a range of distinct problems - for example, during the course of this project, BEAST has featured in two widely reported biological discoveries. In 2013, a new species of river dolphin was discovered in the Araguaia river of Brazil - the first discovery of a new river species since 1918. BEAST was used to reconstruct the evolutionary relationship between this species and other dolphin species, and to infer when this species diverged from the other species of river dolphin found in Brazil [29]. Another publication in 2013 described the discovery of the oldest yet sequenced DNA - between 560 and 780 thousand years old - from a horse bone retrieved from permafrost. In this study, BEAST was used to reconstruct the population history of species in the genus *Equus*, taking this new data into account [30].

BEAST is also a proven tool for carrying out evolutionary analysis of a wide variety of infectious diseases and viral pathogens. Notable achievements of Bayesian phylogenetic analysis with BEAST include origin analysis of the 1998 Al-Fateh Hospital HIV outbreak, detection of transmission clusters in the UK HIV epidemic, and inferences about the origin of HIV-1 on a global scale [11, 12, 13, 14]. More recently, BEAST has been to analyse emerging epidemics in real time, such as the Middle East respiratory syndrome coronavirus (MERS-CoV) and the 2009 H1N1 influenza A pandemic [38, 16, 17]. This project will focus on the application of BEAST in these contexts, but it should be noted that BEAST is not restricted to these types of analyses, and several tools and methods developed as part of this project may have use outside the context.

2.3.1 Evolutionary parameters of interest

2.3.1.1 The phylogeny

A phylogeny represents information about the inferred evolutionary relationships between a set of observed entities, situated at the tips (leaves) of the tree. The horizontal dimension of the phylogeny shown in Figure 2.2 represents amount of genetic change - longer branches in the phylogeny indicate more genetic change, with branch lengths usually measured in number of nucleotide substitutions (for example, a mutation from A to a G) in the genetic code divided by the length sequence length. Each leaf node of the tree represents an observed genomic sequence, and the points at which branches merge can be considered internal nodes representing hypothetical, unobserved ancestral states. The numbers associated with each of

thee nodes correspond to the support for the node, which are usually stated as posterior probabilities ranging from 0 to 1. While a phylogeny is an intuitive way to view evolutionary relationships, the parameters estimated along with the phylogeny are often of greater import.

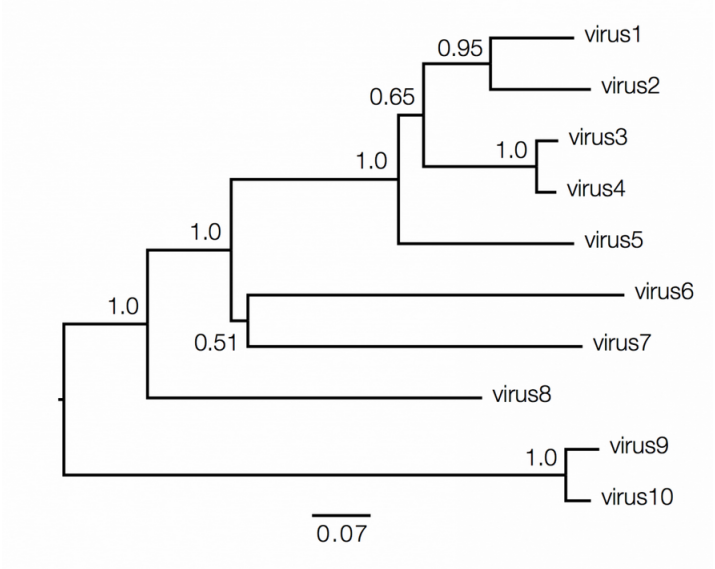
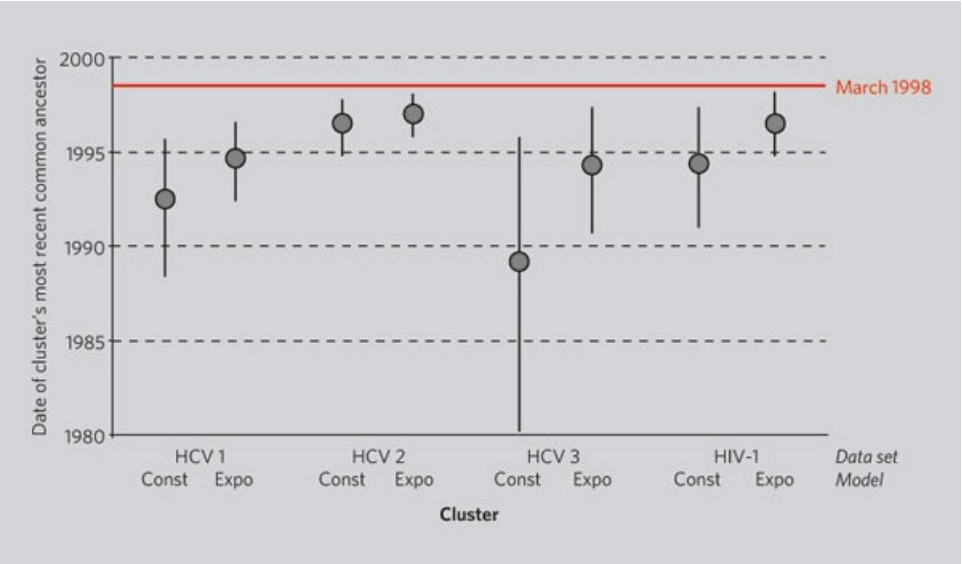


Figure 2.2: An example phylogeny

2.3.1.2 Time to the most recent common ancestor

The root node of the phylogeny represents the (hypothetical) most recent entity from which all observed viral strains represented in the phylogeny are descended, or their most recent common ancestor (MRCA). As stated above, the horizon dimension of a phylogeny represents genetic change, but this can be converted into time, given that the relationship between genetic change and time is clearly defined, and such a relationship is defined by a molecular clock assumption (discussed below). It therefore becomes possible to discuss the time to the MRCA, or TMRCA, and this gives the time of origin of a pathogen - an essential parameter for retracing the origin of a virus.

For example, the largest documented incident of hospital-induced HIV infections occurred in 1998, where over 400 children were infected with human immunodeficiency virus type 1 (HIV-1) and hepatitis C virus (HCV) at the Al-Fateh Hospital in Benghazi, Libya. This epidemic resulted in an even larger international outcry when six foreign medics working at Al-Fateh were accused - by the Libyan government - of intentionally infecting the children, and sentenced to death [10]. A study used BEAST and all available genomic sequences of the HCV and HIV pathogens to infer the TMRCA of the outbreak under different evolutionary models. These results are illustrated in Figure 2.3, with vertical lines representing the 95 percent highest posterior density intervals of each TMRCA estimate, and the red line showing the time of arrival of foreign staff at Al-Fateh.



Reproduced from, Oliveira et al, *Molecular Epidemiology: HIV-1 and HCV sequences from Libyan outbreak*, Nature 444, 836-837.

Figure 2.3: TMRCA estimates given by BEAST on Al-Fateh data compared to arrival of 6 foreign medical staff at the hospital used as evidence for the innocence of the staff

2.3.1.3 Basic reproductive rate and intrinsic growth rate

A BEAST analysis assumes prior growth model for the virus - normally either logarithmic or exponential - and the intrinsic growth rate, r_0, is a hyperparameter (a parameter of a prior) of this distribution. The value of r_0 can be inferred, and this estimate can be used to calculate the basic reproductive rate, R_0, which was introduced in Section 1.2.1, and is defined as the average number of secondary infections caused by one infection by a viral pathogen.

The estimate of R_0 is crucial, as it is directly involved in the definition of a potential epidemic. If $R_0 > 1$, a pathogen has the potential to spread throughout a population, causing an epidemic, but if $R_0 = 1$, the pathogen is expected to become endemic in a population. It follows that if $R_0 < 1$, a pathogen will die out over time. Clearly, the higher R_0, the most severe an outbreak, and the value of R_0 also allows ranking of the severity of an outbreak relative to diseases with known R_0, such as Smallpox (see table 1). The transformation of r_0 to R_0 is dependent on pathogen-specific assumptions on generation time and population, but in all cases, if r_0 is greater than 0, R_0 will always be bigger than 1 under this transform, implying the least that the virus has at least endemic potential.

2.3.1.4 Evolutionary rate

A majority of viral pathogens are characterised by a high rate of evolution - HIV being a notorious example, as its rapid evolution makes it resistant to immune response and anti-viral drugs. This high rate of evolution in viruses is largely due to a faster rate of reproduction relative to their hosts. Change s in the genomic

Values of R_0 of well-known infectious diseases

Disease	Transmission	R_0
Measles	Airborne	12–18
Pertussis	Airborne droplet	12–17
Diphtheria	Saliva	6–7
Smallpox	Social contact	5–7
Polio	Fecal-oral route	5–7
Rubella	Airborne droplet	5–7
Mumps	Airborne droplet	4–7
HIV/AIDS	Sexual contact	2–5
SARS	Airborne droplet	2–5
Influenza (1918 pandemic strain)	Airborne droplet	2–3

Adapted from http://en.wikipedia.org/wiki/Basic_reproduction_number, available in the public domain.

Figure 2.4: R_0 values for notable infection diseases

sequences of pathogens come about either through mutations - errors during the replication of genetic material, particularly common in viruses, resulting in a change of base at one or more sites in the sequence - and substitutions, which come about as a result of natural selection or random genetic drift [25]. BEAST infers the substitution rate as the rate of evolution of a virus, and reports this as a number of substitutions per site per year of evolution. The evolutionary rate has been estimated for many viruses (see [40]).

2.4 Modelling sequence evolution

Sections 2.4.1 to 2.4.3 introduce a subset of components of the evolutionary model used by BEAST to perform inference. This model of evolution has been built over many years, through principled research and experimentation, and this introduction serves only to impart to the reader a basic intuition regarding how the specification of certain parts of the model affect estimates given for evolutionary parameters of interest. It also serves as an introduction to the wide range of parameters the priors that can be specified when specifying a BEAST analysis For a deeper discussion of the evolutionary model, please refer to the BEAST paper[7].

2.4.1 The molecular clock

BEAST carries out inferences using rooted trees that are tightly coupled with a time scale. The length of each branch corresponds to genetic change, such that, given a branch length μ between the ancestral sequence A and one of it's descendants D, each site in A is expected to undergo μ nucleotide substitutions by the time it evolves to D.

Given that the dates on which viral isolates are sampled are known, it is possible

to determine the number of decimal years between two sequences, and represent a branch length using time differences. Because some isolates, or more generally species, evolve at faster rates than others, the μ branch length measure may not be directly proportional to this time measure. An evolutionary model that employs a strict molecular clock imposes the constraint that the expected number of substitutions per year, μ, is independent of which isolate's evolution is being considered. The strict molecular clock becomes more unrealistic as the period of evolution increases - for example, consider observing evolution between species where one breeds much faster than the other.

The relaxed molecular clock assumption overcomes this simplification by allowing the rates of evolution to vary along all branches of a phylogeny. BEAST differentiates itself from other Bayesian phylogenetic software packages, such as MrBayes [41], as it models phylogenies on a time scale, which allows for relaxed phylogenetics - phylogenetic analysis with a relaxed molecular clock. Such analysis has been shown to be more precise and accurate in estimating phylogenetic relationships, and, most relevantly to this project, actually allows inference of parameters such as TMRCA, which is of direct importance in understanding pathogen outbreaks [28]. The ability to perform relaxed phylogenetics is one of the major reasons that BEAST is now the state of the art tool for phylogenetic inference [7].

2.4.2 Substitution models

A substitution model aims to model molecular evolution (see Section 2.3.2.4) by describing the process by which one sequence changes to another through sustained nucleotide substitutions, and the assumptions made in the substitution model strongly affect estimates of the evolutionary rate. The most popular substitution models make the following assumptions: a substitution at one site does not affect the probability of a substitution at another site (independence) , natural selection does not act on the substitutions (neutrality), and a single site can undergo multiple substitutions (finite sites). Such models are time-reversible (ie an ancestral sequence can be recovered by reversing the substitutions that accumulated to make a descendent), and can be concisely modelled as continuous-time Markov chains.

Markov substitution models of sequence evolution are parametrised by a 4 by 4 rate matrix Q, where Q_{ij} gives the rate at which bases of type i change to type j, and an equilibrium vector of base frequencies π. The rate matrix is used to compute a transition matrix function, $P(t)$, which maps branch lengths (in substitutions over time) to a matrix of conditional probabilities, where $P_{ij}(t)$ represents the probability that, after time t, base j appears at a certain nucleotide position, given that base i was at that position at time 0.

A large number of substitution models exist, ranging from the simplest - the JC69 model [26] where Q is a symmetric matrix with off-diagonal elements all equal to the overall substitution rate μ divided by 4 - to more complex models that include the K80 model [27] , which distinguishes between more commonly observed substitutions due to the biological properties of base pairs.

An additional layer of complexity is introduced when considering the fact that different parts of a DNA sequence evolve at different rates - for example, some areas that encode fundamental parts of a genome that are directly necessary to the life of an organism may not change at all or be highly conserved. Ares that do not change are known as invariant sites, and a discretised gamma distribution is used to model among-site rate variation (ie slow to fast evolving regions), parametrised by a gamma shape parameter and rate parameter.

2.4.3 Tree priors

Coalescent theory is concerned with models that trace genetic history from observed organisms to the most recent common ancestor, and such models also incorporate information on demographic history of pathogens to model how the population of a pathogen changes over time. BEAST allows population size to be modelled by either a constant growth, exponential growth, or logistic growth model. These models serve as priors on the age of nodes in the phylogeny, and the growth rate is a hyperparameter of this prior. Hence, the choice of of model strongly influences estimates of the growth rate of a virus.

The tree prior includes a variety of additional parameters that model branching times, constraints on specific topological features (which may model prior knowledge gleaned through fossils), and priors over the date of each node in the tree. Clearly, inference of a phylogeny is dependent not only on these priors, but on other aspects of the evolutionary model discussed above.

2.5 Inference with BEAST

Section 2.4 introduced the key parameters required for modelling molecular sequence evolution - the substitution model parameters μ, the gamma shape parameter for modelling rate variation along sequences α, possible tree topologies τ, and branch lengths with a clock assumption, β. Given these parameters it is possible to define a statistical model in a phylogenetic context which can be used to carry out inference.

2.5.1 Metropolis-Hastings Markov Chain Monte Carlo

In the following discussion, the reader is assumed to have an understanding of probability theory and Bayesian statistics. Let $\theta = \{\tau, \beta, \mu, \alpha\}$, which represents a particular tree consisting of a specific combination of branch lengths, gamma shape parameter, and substitution parameters. Estimating the joint posterior probability distribution of these parameters given a set of sequence data is the key inference task carried out by BEAST. More formally, if X denotes a set of observed sequence data, $P(\theta|X)$ is inferred. By Bayes' theorem:

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)} \tag{2.1}$$

The likelihood function, $P(X|\theta)$ for a particular tree t_k is integrated over all possible substitution parameters and branch lengths, ie:

$$P(X|\tau_k) = \int_{\beta_i} \int_{\mu} \int_{\alpha} P(X|\tau_k, \beta_i, \mu, \alpha) P(\mu) P(\alpha) d_{\beta_i} d_{\mu} d_{\alpha} \tag{2.2}$$

These high dimension integrals required for computing the posterior cannot be solved analytically for almost all practical applications, and hence the posterior must be approximated with Metropolis-Hastings MCMC. MCMC is a algorithm that allows for efficient sampling from a posterior distribution by defining a Markov chain over a state space of parameters of the posterior, in this case θ. The stationary distribution given by this chain is the posterior distribution of parameters that the algorithm aims to draw samples from.

The MCMC algorithm begins this random walk by starting the chain at a specific point in the parameter space, defined by θ_{start}. At each state in the walk, the parameter values of that state are sampled, a proposal distribution is used to propose a new state θ_{new} given the current state $\theta_{current}$. The algorithm will jump to this new state with a probability proportional to the relative probability of these states under the posterior. The length of the MCMC procedure is defined as the chain length or number of sampling steps. Given a large enough chain length, samples from the chain will become valid samples from the posterior distribution, and the posterior mean of samples over the total chain length can be used as inferred values of parameters of interest.

Mixing, burn-in and convergence

At the start of a MCMC procedure (referred to as a 'run'), θ_{start} would often consist of a random starting tree and arbitrarily chosen branch lengths - a set of parameters with low likelihood under the model. As the chain progresses, the likelihood will increase significantly as the sampler moves towards regions of high posterior probability (given the nature of proposals of for new states). This initial phase of a run is termed burn-in and samples from this period are not used in posterior mean estimation of parameters, as they heavily dependent on the (randomly chosen) starting point of the chain. For most usages of BEAST in the literature, it is common for researchers to discard at most the first 10 percent of samples from a chain as burn-in.

A chain's mixing rate is defined as how quickly the chain samples from the main regions of the posterior - a higher mixing rate is desirable as this means adequate or high quality samples are being drawn from the posterior at a faster rate [6]. The time it takes for a chain to mix can be adjusted by tuning the proposal distributions for certain model parameters. The quality of samples is strongly linked to whether the algorithm has converged or not.

Determining convergence

Determining the convergence of the run is a problem with no known deterministic solution. This is an important problem in WILDEBEAST, as a crucial component of a real time characterisation system for pathogens is the ability of the system to give parameter estimates even when a BEAST run has not yet completed its chain length. WILDEBEAST should be able to inform a user of the quality of estimates from a run with respect to mixing rate, and make decisions based on this.

It is important to remember that unlike other techniques such as the ExpectationâĂŞmaximization algorithm, MCMC must not only reach a high probability region, but sample from it adequately to so that posterior mean estimates from samples give good approximations for parameters. The likelihood of samples can be monitored to determine convergence by inspecting whether likelihood values have stabilised over samples, but continue to sample from the full distribution. The BEAST software package includes the tool Tracer, which provides a GUI for visual inspection of likelihood of a run, and this is often used by evolutionary epidemiologists to evaluate convergence.

Independent runs of the same analysis are also a popular way of monitoring convergence in BEAST. However, this method comes at a high computational cost. The tree topology is often the most difficult parameter to draw samples from, and researchers often inspect the variance between sampled trees of different runs to determine convergence [6].

The above convergence monitoring methods require manual intervention, while measuring effective sample sizes (ESS) for each parameter in a chain is a more automatic method of monitoring mixing behaviour and convergence, and hence was chosen to be the main method for measuring quality of samples in the automated processes implemented by WILDEBEAST. The ESS of a parameter is defined as the number of independent samples of it that have been drawn from the posterior over the chain length so far. This is computed by taking the chain length, without the burn-in period, and dividing it by the auto-correlation time (ACT). ACT gives an intuition of how uncorrelated samples from the posterior are in a chain, and makes use of the average number of states between two samples for them to be uncorrelated [6]. A higher ESS for one parameter estimates compared to another reflects that it is a better approximation, as a estimate with higher ESS is computed from a larger number of independent samples.

2.6 Related work

2.6.1 Phylodynamics

The importance of phylogentic tools such as BEAST towards the study of epidemics is highlighted by growing interest in the area of phylodynamics [8]. This field aims to meld understanding of the evolution of pathogens, using phylogenetic tools such as BEAST, with epidemiological analysis of a pathogen, and such joint

analysis is critical towards effective understanding and control of epidemics through policy [9, 14]. This project will only focus on characterisation of pathogens through phylogenetic techniques, though it should be noted that traditional epidemiological analysis use distinct approaches, such as case studies, that are not usually informed by analysis of molecular sequence data.

2.6.2 Forecasting

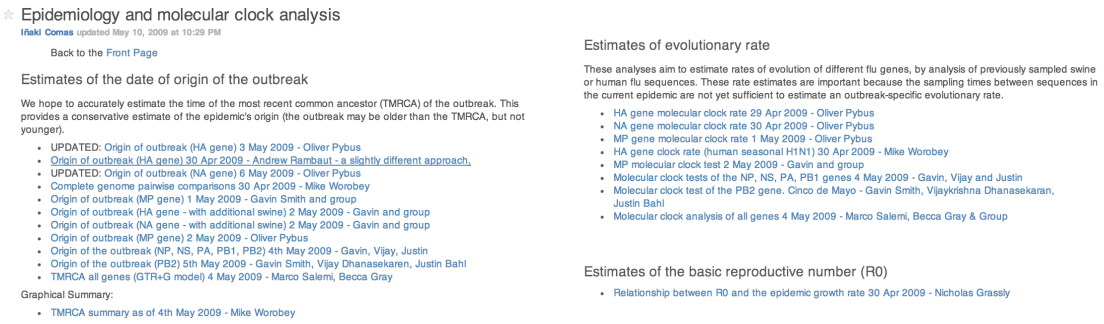
Recognising a novel outbreak of a pathogen is a non-trivial task, and is a prerequisite for pathogen characterisation. In certain cases, pathogens have circulated for years in human populations without detection - HIV being an example. There are a number of existing systems for forecasting epidemics, such as surveillance systems employed for influenza by the Centers for Disease Control and Prevention (CDC), a public health institution headquarters in the USA. Approaches to this task must be carried out in real time, and novel methods for detecting outbreaks of influenza have been proposed in systems such as Google Flu Trends, which was built to predict influenza reports given by the CDC. Google Flu Trends also offers a web-based tool that gives real time information on flu activity on a global scale by analysing user search terms [37]. A study showed that this system could detect outbreaks of other pathogens, such as West Nile virus, though results given by this tool have been strongly criticised by the scientific community as results were shown to over predict CDC estimates on flu prevalence by more than 50 percent [39]. Tools like Google Flu Trends clearly have limited scope as data can only be gathered from countries where Google Search is actively used [37].

2.6.3 Real time characterisation of epidemics

Once a pathogen is identified, quantifying its potential to cause epidemics and even pandemics is an essential step that requires systematic investigation of data. Such data is usually limited in the early stages of an outbreak.

The WHO Rapid Pandemic Assessment Collaboration report used three distinct epidemiological analyses to compute the R_0 of H1N12009 just two months after the WHO announced issued a global pandemic alert for the outbreak. The first analysis used estimates of the start date, cumulative number of infections, and the generation time of the virus (using prior knowledge from other viruses) to infer this quantity. Bayesian coalescent population genetic analysis was also carried out to give a second estimate. A third method fitted epidemic models to the dynamics of the pathogen observed in a well defined setting - the La Gloria outbreak that had occurred earlier in the year in Mexico. Overall, estimates given with these methods corroborated with those given through BEAST analysis of the available sequence data. The estimated values of R_0 were found to be significantly higher of those found for seasonal flu, but not as serve as those seen in past influenza pandemics, such as the 1918 pandemic [17].

A key study that has served as an inspiration for this project investigated how well the three parameters of interest could be estimated with BEAST in a retrospective


Epidemiology and molecular clock analysis

Iñaki Comas updated May 10, 2009 at 10:29 PM

Back to the Front Page

Estimates of the date of origin of the outbreak

We hope to accurately estimate the time of the most recent common ancestor (TMRCA) of the outbreak. This provides a conservative estimate of the epidemic's origin (the outbreak may be older than the TMRCA, but not younger).

- UPDATED: Origin of outbreak (HA gene) 3 May 2009 - Oliver Pybus
- Origin of outbreak (HA gene) 30 Apr 2009 - Andrew Rambaut - a slightly different approach.
- UPDATED: Origin of outbreak (NA gene) 6 May 2009 - Oliver Pybus
- Complete genome pairwise comparisons 30 Apr 2009 - Mike Worobey
- Origin of outbreak (MP gene) 1 May 2009 - Gavin Smith and group
- Origin of outbreak (HA gene - with additional swine) 2 May 2009 - Gavin and group
- Origin of outbreak (NA gene - with additional swine) 2 May 2009 - Gavin and group
- Origin of outbreak (MP gene) 2 May 2009 - Oliver Pybus
- Origin of the outbreak (NP, NS, PA, PB1, PB2) 4th May 2009 - Gavin, Vijay, Justin
- Origin of the outbreak (PB2) 5th May 2009 - Gavin Smith, Vijay Dhanasekaren, Justin Bahl
- TMRCA all genes (GTR+G model) 4 May 2009 - Marco Salemi, Becca Gray

Graphical Summary:

- TMRCA summary as of 4th May 2009 - Mike Worobey

Estimates of evolutionary rate

These analyses aim to estimate rates of evolution of different flu genes, by analysis of previously sampled swine or human flu sequences. These rate estimates are important because the sampling times between sequences in the current epidemic are not yet sufficient to estimate an outbreak-specific evolutionary rate.

- HA gene molecular clock rate 29 Apr 2009 - Oliver Pybus
- NA gene molecular clock rate 30 Apr 2009 - Oliver Pybus
- MP gene molecular clock rate 1 May 2009 - Oliver Pybus
- HA gene clock rate (human seasonal H1N1) 30 Apr 2009 - Mike Worobey
- MP molecular clock test 2 May 2009 - Gavin and group
- Molecular clock tests of the NP, NS, PA, PB1 genes 4 May 2009 - Gavin, Vijay and Justin
- Molecular clock test of the PB2 gene. Cinco de Mayo - Gavin Smith, Vijaykrishna Dhanasekaran, Justin Bahl
- Molecular clock analysis of all genes 4 May 2009 - Marco Salemi, Becca Gray & Group

Estimates of the basic reproductive number (R0)

- Relationship between R0 and the epidemic growth rate 30 Apr 2009 - Nicholas Grassly

Figure 2.5: Manual characterisation efforts for H1N1, replicated from `http://tree.bio.ed.ac.uk/wiki/projects/influenza/HumanSwine_H1N1_Influenza.html`

study. Cumulative sequence data was created for each month of 2009, starting in April, and BEAST runs carried out on each dataset. It was shown that the TMRCA converged to estimates of 2 Februrary 2009, 0.00393 substations/site/year for the rate, and a R_0 of 1.12. The logistic growth model was found to be a better fit in later stages of the epidemic, and the study concluded that analysis of sequence data could make earlier parameter estimate feasible in future epidemics.

2.6.4 Systems for real time characterisation

Manual systems

While that above studies show that BEAST can be used effectively for real time characterisation, no system exists for carrying out this procedure. BEAST has been built as a desktop tool to be used by individuals in the analysis of fixed datasets, and its initial implementation does not cater well to the environment imposed by epidemics.

Previous real time characterisation systems are websites that serve mostly as a place for researchers to share, document, and discuss new sequence data and the results of analysis carried out individually on data retrieved from external repositories. For example, Figure 2.5 shows a screenshot of a webpage maintained by the Institute of Evolutionary Biology group at the University of Edinburgh, where researchers posted updates parameter estimates given by analysis performed over H1N12009.

This webpage was later adapted for use in other epidemics, and it available at `http://epidemic.bio.ed.ac.uk/`). The central aim of the website is for presenting new results as more data becomes available, and this system still remains a blog-styled website where researchers share and discuss results.

Websites that host short peer reviewed articles have also been introduced to allow rapid communication of research results during epidemics. An example of such a system is PLOS Currents:Outbreaks (available at `http://currents.plos.org/outbreaks/`, where researchers can submit characterisations of pathogens for publication under a quicker peer review system. However, such a system is lim-

ited as simple re-estimates of parameters re unlikely to be published, content is manually created and varies, and a time-lag still exists for peer review.

Automated webservices

Bioinformatics webservices have become increasingly popular, and are an important component of TBI. For example, the Cornell BRC Bioinformatics Facility hosts a webservice that allows analysis of biological data with a wide range of common bioinformatics tools that include BEAST. DataMonkey (`http://www.datamonkey.org/`) is another popular webservice that allows researchers from the general public to analyse their uploaded sequence data with a variety of statistical sequence evolution algorithms.

These webservices only offer automation in a limited capacity, for example, the Cornell tool will automatically notify you once your BEAST job is complete. None of these systems have been built for specific automated analysis during epidemics, and do not deal with any of the complexities inherent in analysis epidemics using MCMC, nor seek to publicly display estimates of parameters whilst allowing a number of researchers to interact with analysis and sequence data. A system that allows this would be a novel tool in the field of molecular epidemiology.

Chapter 3

WILDEBEAST interface

Section 3.1 sets forth the motivations for a web-interface. The WILDEBEAST interface is then described in Section 3.2 through a step-by-step introduction to adding, monitoring, and managing an epidemic. The user-facing elements of each section of the service are defined. The reader is encouraged to visit the WILDEBEAST website at `http://kimura.bio.ed.ac.uk:8080/WILDEBEAST/index`.

3.1 Motivation

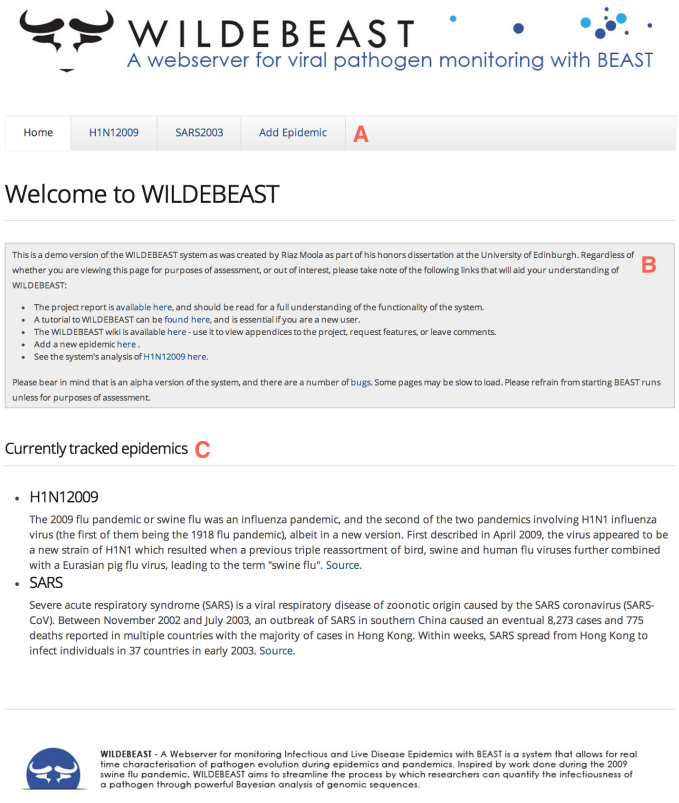
A web service is inspired by previous attempts at constructing websites to facilitate early characterisation of outbreaks (See Section 2.6). Previous attempts have served mostly as a static pages on which researchers collaborate. The web interface component aims to provide a framework that both simplifies and enhances these collaborations by providing a platform for researchers to directly view and manage BEAST analyses, data, and statistics about the timeline of an epidemic. A best knowledge page automatically factors in the results of these analyses to summarise estimates of important evolutionary parameters.

Another motivation is that there is no publicly accessible system in place for automatically tracking emerging epidemics in real time. Current attempts, reviewed in Section 2.6, suffer from a lack of automation. For example, these systems rely entirely on manual human retrieval and analysis of sequence data, and sporadic posts or publications to give updated estimates of parameters. All of these processes introduce a significant time lag from the availability of new sequence data to an update on evolutionary estimates.

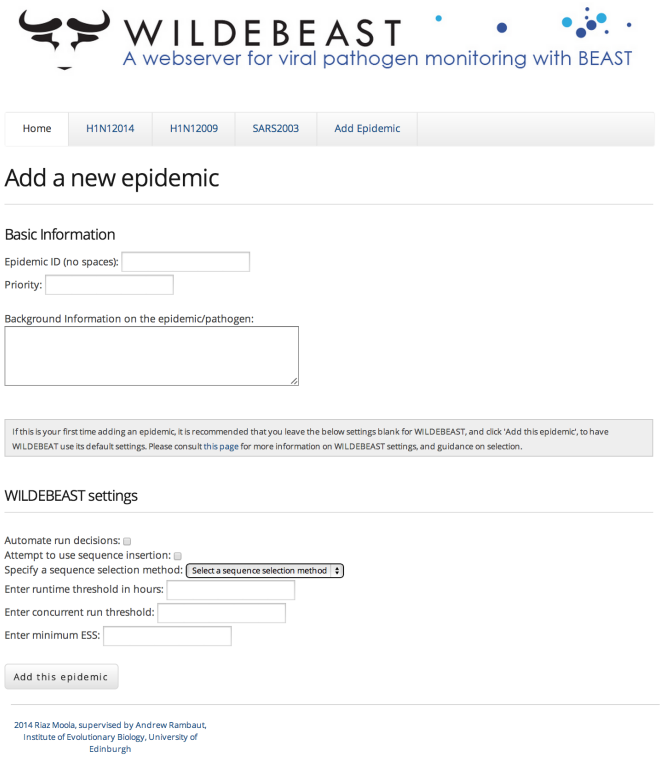
3.2 How to use WILDEBEAST

3.2.1 Adding an epidemic

This section presents a guide of how to interact with WILDEBEAST to monitor a hypothetical influenza A outbreak occurring in in 2014. Figure 3.1a shows the main portal of WILDEBEAST, which can be accessed by visiting `http://kimura.bio.`



(a) Welcome portal



(b) Add epidemic

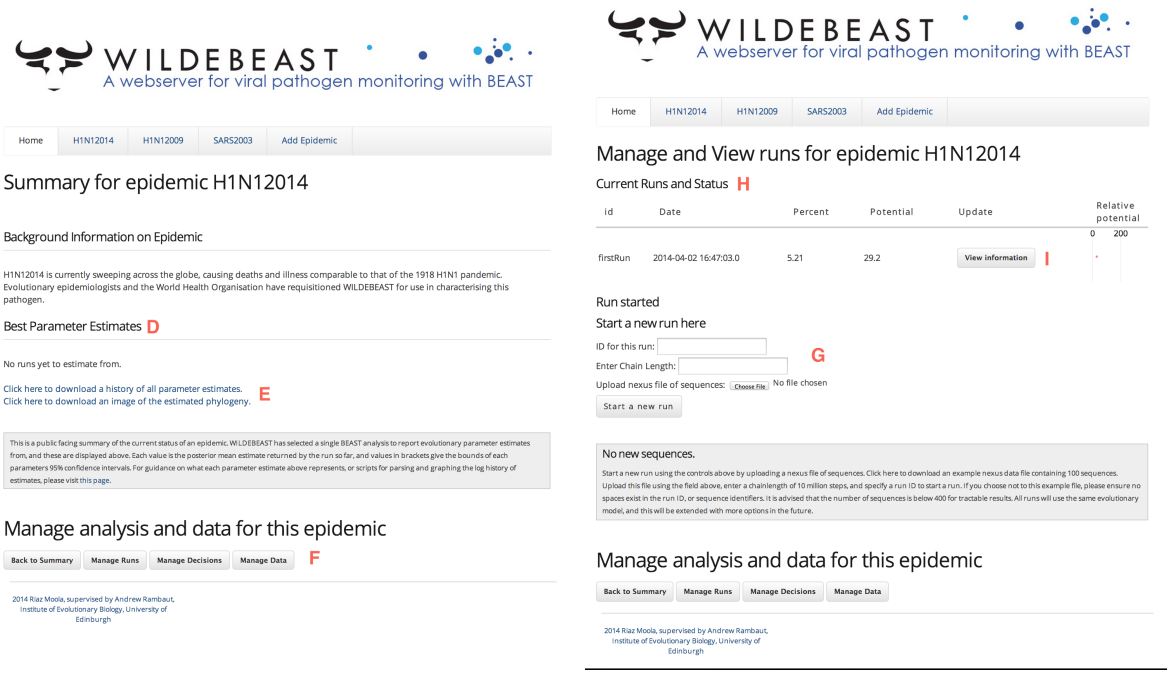
Figure 3.1: Setting up H1N12014

`ed.ac.uk:8080/WILDEBEAST/index`. This page opens to a welcome portal (Figure 3.1a) which first provides some basic information about the system with links to a wiki of tutorials (B), a section listing information on all currently tracked epidemics (C), and a navigation menu to explore epidemic-specific pages (A) ordered by importance .

To tell WILDEBEAST about this new epidemic, click the Add Epidemic tab on the top menu. The page shown in Figure 3.1b will open. Enter the ID 'H1N12014' into the Epidemic ID field and follow the guidance on this page to fill the other fields. Do not fill out the operational settings section (discussed in detail in Chapter 6) to have WILDEBEAST use the defaults, and click 'Add this epidemic'. You will be returned to the homepage.

3.2.2 Manually starting a BEAST analysis

Each epidemic has four main webpage displays - a summary page, runs page, decisions page, and data page. To access these displays for your new epidemic, click the newly added item to the menu bar to navigate the summary page for H1N12014. The page shown in Figure 3.2a will open, where estimates of evolutionary parameters are reported (D), and a log file of all estimates over time can be downloaded (E). WILDEBEAST reports no estimates because no BEAST analysis



(a) Epidemic summary page **(b)** BEAST analysis/Runs page

Figure 3.2: Starting an analysis

have been initiated. Click the 'Manage Runs' button under the navigation menu (F) to be taken to the runs page, which will look similar to Figure 3.2b. Use this page to create a new run by following the instructions shown in the box below the 'Start a new run' section, which will provide example sequence data (G).

After creation of a new run, this page will look like Figure 3.2b. The runs table (H) is dynamically generated with the Javascript library $D3.js$, and summarises the status of all BEAST analysis carried out either manually or automatically by WILDEBEAST for this epidemic. Runs are ordered by the potential assigned to each by the system, based how well the data for a run represents the evolution of a pathogen (see Chapter 7).

Click the 'View Information' button (I) to open up the Run information display (Figures 3.3a and 3.3b), on which you can view up to date information on the dataset, status, estimates, and health (J) of a particular run. All run files generated by BEAST can be downloaded, and the run can be stopped or deleted (K). Stopped runs are still visible through the interface, and estimates given by them factored into the decision processes, but deleted runs are completely removed from the system. Close this window, and click on the 'H1N120014' menu item to return to the summary page to notice that WILDEBEAST now reports parameter estimates for the virus from the run you just started, as it is the only available analysis for the epidemic.

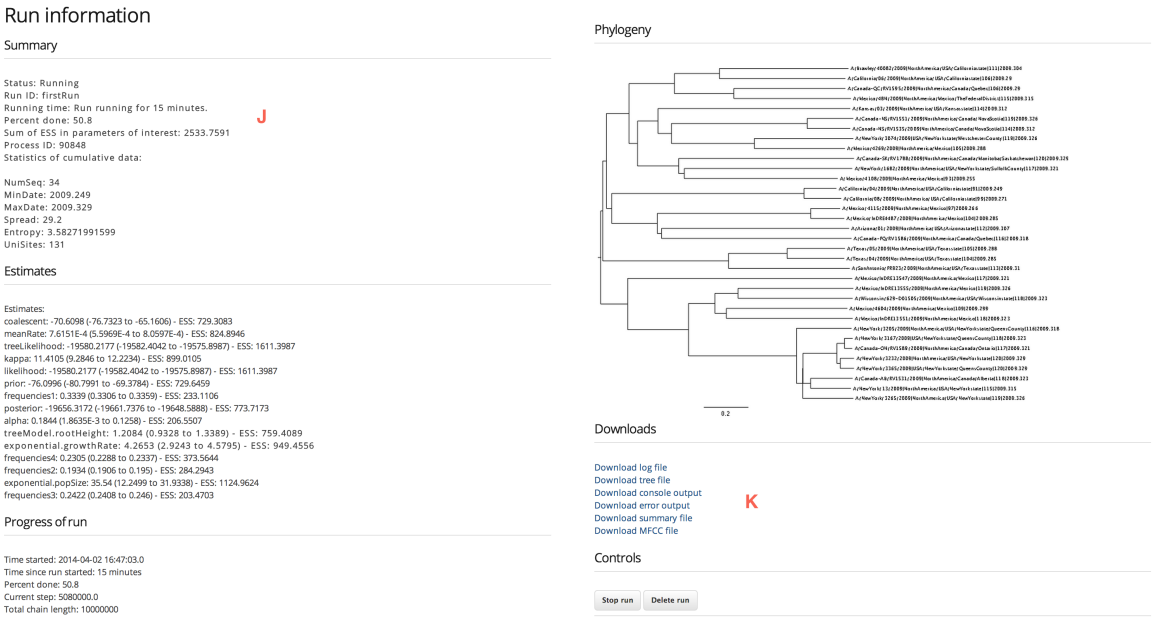


Figure 3.3: Run-specific information and control page

3.2.3 Viewing data for an epidemic

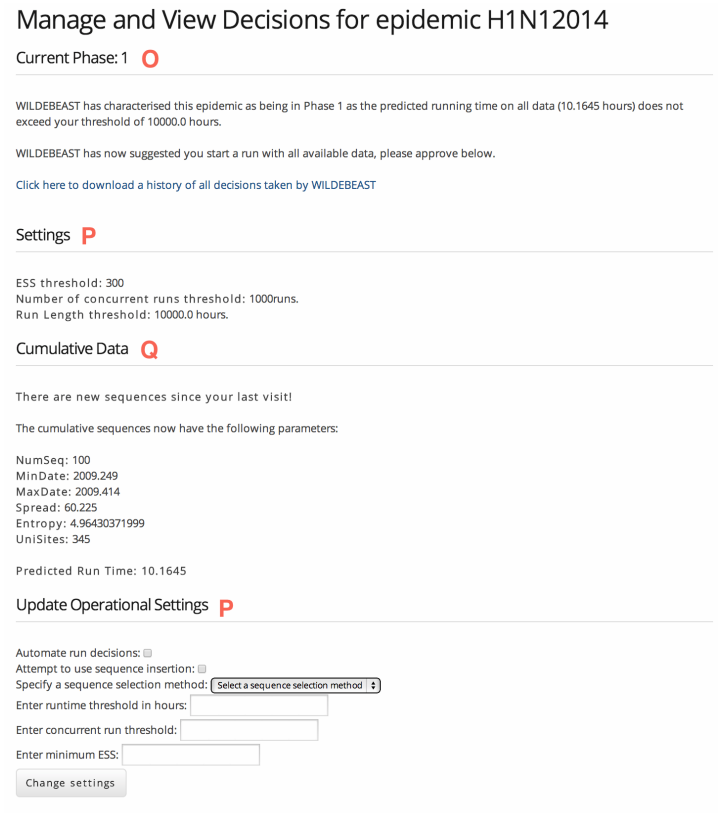
Click the 'Manage Data' button and the page shown in Figure 3.4a will open. The locations of where sequences in the cumulative sequence knowledge for the epidemic (currently only the set of sequences used in the first run) were sampled from are plotted using a hierarchical clustering method described in Chapter 5 (L). This map is important in revealing the extent of the pathogen. A list of all sequence identifiers and their information is provided below (M), as well as a section where new sequences can be manually uploaded (N).

3.2.4 Autonomous functions of WILDEBEAST

As new data arrives during an epidemic, WILDEBEAST carries out several decision processes, which are fully described in the following chapters. The 'Manually add new sequences' section of the data page allows a user to manually notify WILDEBEAST of new sequences, by uploading a set of sequences to a specific location on the server where WILDEBEAST continually checks for new data. This project does not focus on the retrieval of new sequences, but WILDEBEAST can easily be extended to automatically search public sequence repositories for sequences.

Follow the instructions under the this section to download and notify WILDEBEAST of the arrival of new sequences in the H1N12014 epidemic. Then, click the 'Decisions page' button to display the page shown in Figure 3.4b. This page shows information described in detail in Chapter 7 such as the phase H1N12014 has been categorised as beining in with a downloadable log of all actions taken by the system (O), current operational settings and an interface to update them (P), and features of the cumulative sequence set, included predicted runtime (Q). Notice

(a) Manage data page



(b) Manage decisions page

Figure 3.4: Data and decision tracking

that the system is aware of new data - soon, WILDEBEAST will start an automated run on the data you just uploaded.

WILDEBEAST is scalable as it allows concurrent characterisation of multiple epidemics. Use the main header to navigate to the ‘H1N12009’ epidemic, which has been set up with a demo version displaying an epidemic that has progressed much further. All pages are dynamically generated using a number of databases, Java servlets, and other tools that run on a webserver. Chapter 4 discusses the architecture behind these displays in detail.

Chapter 4

Architecture

The WILDEBEAST service was implemented using Java Servlets that run in the Apache Tomcat servlet container. 24 Java classes, a set of MySQL databases, the BEAST software package, and a number of shell and Python scripts were written to comprehensively define the implementation. Java classes are divided into three packages - $corejobs$, $datajobs$ and $pageviews$. The service is currently implemented on the e Kimura server at the Institute of Evolutionary Biology, University of Edinburgh.

4.1 The $corejobs$ package

The main functionality implemented by the 8 $corejobs$ classes are presented in this section. $corejobs$ classes implement functions that interface directly with the BEAST pipeline, carry out decision making, and log the system state.

BEAST software pipeline

The BEAST software package (version 1.7.5) already implements a comprehensive suite of Java classes which function in a pipeline (either through command line calls or a desktop graphical user interface) in order to carry out a BEAST analysis. An overview of the process is as follows:

1. The genomic data to be analysed are specified in either Nexus or FASTA format (see Figure 4.1 for an example). Both formats are commonly used in bioinformatics to specify a set of aligned genetic sequences. In the context of this project, each sequence corresponds to the genetic information sequenced from an isolate - a viral sample taken from an individual or organism.
2. The FASTA or Nexus file is transformed into a BEAST XML file which fully defines the BEAST analysis that is to be performed, including parameters of the evolutionary model, prior assumptions, and length of the MCMC chain.
3. The BEAST XML file is used as input to a separate Java application which

begins the MCMC. This process continually produces console output, a log file, and trees file as it runs. Each line of the log file records the sample from the posterior for each parameter at a step of the chain, and the trees file continually logs sampled trees specified in newick format. No output is produced during burn-in.

```
>A/California/04/2009|NorthAmerica/USA/Californiastate|91|2009.249
ATGGAGAGAATAAAAGAACTGAGAGATCTAATGTCGCAGTCCCGCACTCGCGAGATACTCACTAAGA
>A/Mexico/4108/2009|NorthAmerica/Mexico|93|2009.255
ATGGAGAGAATAAAAGAACTGAGAGATCTAATGTCGCAGTCCCGCACTCGCGAGATACTCACTAAGA
```

Figure 4.1: Truncated example of the FASTA representation of genetic sequences from two isolates of H1N12009. The identifier of each sequence follows the > and then next line contains its genetic sequence

BEAST utilities

$BeastGen$ and $LogAnalyser$ are utilities included with BEAST. The classes implemented in $corejobs$ invoke both utilities to create summary statistics of runs which are parsed for the informed that is presented through the web interface, and used in decision making procedures.

$BeastGen$ is a template-driven, command line tool, which uses templates to convert from one format in the BEAST pipeline to another. Templates were provided for converting from FASTA to nexus format, or for converting a FASTA file into a BEAST XML file when combined with an evolutionary model template. Each evolutionary model template fully specifies the setup of the BEAST run, including substitution model, prior son parameters and hyperparameters of the model, and chain length.

As seen in Chapter 2, a single BEAST analysis consists of numerous settings that fully define the evolutionary model. It is outside the scope of this project to investigate the effect of evolutionary model on real time parameter estimates (see [18] for a study on this). Instead, a simpler, computationally tractable evolutionary model that is frequently used for epidemics was specified for all analysis, with only the chain length varying.

The standard evolutionary model and settings template to be used by all analyses is summarised in Table 4.1. Of note is strict molecular clock assumption was used to reduce the run time of analysis significantly, whilst still enabling analysis to give accurate results.

Parameter	**Setting**
Molecular clock assumption	Strict
Growth model	Exponential
Substitution Model	HKY substitution model [42]
Burn-in	10% of total chain length

Table 4.1: Run and evolutionary model parameters

$LogAnalyser$ is another command line utility that helps process the log and tree files generated by a BEAST analysis. $LogAnalyser$ can be invoked to produce a text file that summarises the posterior mean estimates, ESS, and confidence intervals for every parameter from a BEAST log file (which gives the parameters values sampled at each state of a chain so far). The trees log file can be summarised in a similar way to produce a Maximum clade credibility tree (MCC), introduced in Chapter 2.

Learning

A central aim of this project is to build on the web interface to create an intelligent framework for epidemic analysis, and a key component of this is the ability to reason about BEAST runs. MCMC is a stochastic algorithm, and there is no method to deterministically give the run time and ESS for the parameters of a run beforehand. A key feature of WILDEBEAST is the ability to predict run time and ESS given features of a dataset on which an analysis will be run. Predictions can then be used to inform sequence selection and decision making, described below. Chapter 5 explores regression functions learnt for making these predictions, and these functions are implemented in the $corejobs$ classes as a first step of the decision processes.

Sequence selection

The sequence selection task concerns selecting a subset of the cumulative sequence data to run a BEAST analysis on with respect to some constraints. These constraints vary from having an upper limit on the predicted runtime of the analysis (for purposes of reporting) , to attempts to meet or exceed a threshold on the effective samples generated by the Metropolis-Hastings algorithm. Chapter 6 introduces several sequence selection algorithms, and explores the hypothesis that subsets that are more representative of the cumulative sequence dataset for an epidemic with respect to certain parameters of the data, such as timespan, will result in more accurate parameter estimates through BEAST. $corejobs$ classes interface with Python scripts that fully implement these sequence selection algorithms.

Global controller and decision making

A global WILDEBEAST controller is implemented in $corejobs$. This is implemented as a Java CronJob that runs at set intervals to update the status of all runs, check for new data for each epidemic and carry out prediction, sequence selection, run management, logging of evolutionary estimates, and logged of the state of the server. The measures, algorithms, and parameters that define this process are fully described in Chapter 7.

Modularity of design

The previously mentioned templates that are used in conjunction with $BeastGen$ can be used to define a number of specialised evolutionary models. WILDEBEAST can hence be easily extended to different, pathogen-specific, evolutionary models specified by expert evolutionary epidemiologists. This modular approach highlights the ability of WILDEBEAST to characterise future epidemics appropriately.

As WILDEBEAST was built using Java Servlets, the entire application can be packaged as a single Web Application Archive (WAR) file and deployed on any server running Apache Tomcat. The servlets packaged in this WAR interface with a number of external shell and Python scripts. The shell scripts can be modified for different webserver operating systems, and Python scripts extended.

For example, sequence selection and insertion algorithms are implemented as functions in a Python class. Additional selection algorithms can be added to this class without interrupting the operation of WILDEBEAST, allowing seamless integration of new techniques. The version of BEAST run by the server can also be modified in a similar way.

4.2 Storing and monitoring runs: $datajobs$

The $datajobs$ package implements classes that interface with databases, BEAST utilities, and the server file system to monitor the progress of runs, track the cumulative sequence knowledge, and log system estimates, errors and actions to the file system.

The file system

Classes were implemented in the $datajobs$ package to create and manage folders for each epidemic. Subdirectories for each run for that epidemic reside in this folder. These directories contain the BEAST, FASTA sequences, console output, tree, and log files associated with each run. A Runs class was implemented to represent each run in a object-orientated manner. Methods associated with these objects were implemented to interface with $LogAnalyser$ through unix scripts, allowing the system continually generate parameter and MCC summaries for each run, and monitor progress. Sequence maps and phylogenies are also generated for each run and epidemic using these tools and stored in the appropriate folder.

Each run and epidemic is given a unique identifier, enforced through validation of user input. These unique epidemic and run identifiers completely define the paths to all relevant files. For example, the log file of run with identifier $runThird$ under the epidemic $SARS$ has a path of $SARS/runThird/ouput.log$. A logs folder at the top level of the WILDEBEAST application directory stores three text file logs - $estimates.txt$, $errors.txt$, and $actions.txt$. The global controller defines how these are written to.

New sequence data is introduced by the placement of a $newSeqs.fasta$ file in the

top directory of each epidemic folder, as this file is inspected regularly for new sequences by the implemented system controller. While a basic method for modelling the arrival of new data, this inspection process can be extended to a module that actively searches public sequence repositories, such as GenBank, for new sequences relating to a particular viral pathogen. Such a module was considered outside the scope of the project and is discussed in Chapter 9 as a future direction.

Databases

A MySQL database service was implemented to run on the server to assist data management. An $epidemics$ database stores information about all currently tracked epidemics. On creation of a new epidemic by a user, a database with that epidemic ID is generated, which contains a $Sequences$ and $Runs$ table. Sequences and run information are stored in these table with unique identifiers for sequences extracted from the FASTA files. The process ID of every analysis started by WILDEBEAST is stored in the $Runs$ table, allowing a run to be stopped on the server through shell scripts.

Monitoring

Other than the previously mentioned global controller which makes use of $datajob$ classes to update the status of all runs and report parameter estimates, run objects are refreshed every time a user accesses a webpage so that up to date information is always displayed. The percent completion of a run is simply the current step in the chain divided by total chain length, and the time run the difference from the server time when the run was started (logged in the $Runs$ table) from the current server time.

4.3 Webpage displays: $pageviews$

Java HTTP Servlets, Java Server Pages (JSPs), CSS pages, and Javascript scripts are implemented in the $pageviews$ package. 7 JSPs define the graphical user interface, and are dynamically generated through HTTP requests that use the above packages to pass variables into the JSPs which can be viewed by a browser. The Cascading Style Sheets (CSS) that define the look and feel of pages are taken with permission from `http://epidemic.bio.ed.ac.uk/`, and the WILDEBEAST logo and footer were used with the permission of a graphic designer [31].

For the purpose of this project, all pages and functionality are publicly accessible, but a real system would implement a user account system allowing only those with the requisite permissions to view pages beyond the report summary for an epidemic and perform other actions. The summary page for each epidemic is intended to be the public facing page which can be used to inform policy makers. A discussion of individual page displays is omitted as they have been discussed in Chapter 3. Several enhancements to the interface, which is not the main focus of the project, are discussed in Chapter 9.

Chapter 5

Learning

Section 5.1 describes the motivations for a learning component of the system. Section 5.2 describes the process by which training data were generated, and 5.3 explains how prediction rules were learnt from the resulting dataset. A discussion of the evaluation of prediction rules is given in Section 5.3.

5.1 Motivation

While the runtime of BEAST analyses have been rigorously studied with respect to effective sampling of tree space [32], no comprehensive study has been carried out to determine more formally the relationship between features of the data, and expected runtime and effective sample sizes. Carrying out such a study is a requisite step for real time decision making during an epidemic, as it allows the system to plan for timely reports with a measurable level of confidence. In addition, no study has been carried out on how knowledge about previous epidemics could be used to inform future analysis.

5.2 Generating training data

5.2.1 Sequence data

H1N12009

Genomic sequence datasets for H1N12009 were retrieved from a previous real-time study in which an unfiltered and filtered dataset were defined. The unfiltered set consists of all full-genome sequences sampled between April and December 2009, retrieved from the GISAID's EpiFlu database (available at `http://platform.gisaid.org/`). The filtered dataset is defined by only allowing one new sequence isolate to be included into the set per location per day of the epidemic, to try filter out epidemiologically connected cases of the virus [18]. There datasets are stored as FASTA files (see Figure 4.1), with the date a sequence was isolated stored as a decimal year in its identifier. The earliest sequence appears

on March 31st 2009 (2009.249), and the size of the per-week cumulative sequence set for each dataset is plotted in Figure 5.1 for the first 14 weeks of the pandemic. For simplicity, the date of sampling of an isolate is taken as the same as the date of when the DNA of the isolate was sequenced, which is a realistic assumption given the pace at which sequencing technology is increasing.

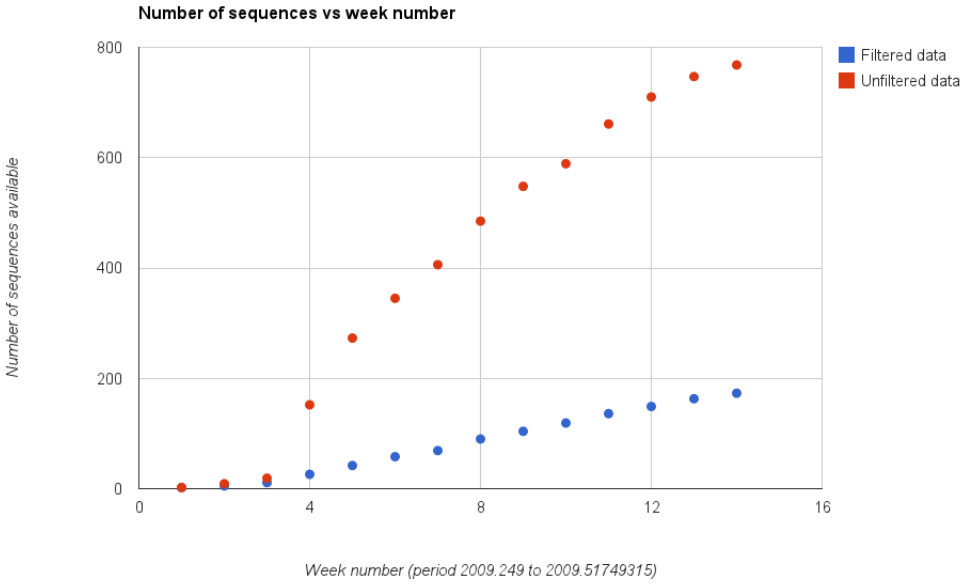


Figure 5.1: Number of sequences in the filtered and unfiltered datasets at weekly intervals over H1N12009

Other epidemics

As the generalisation of WILDEBEAST is an important feature, datasets from other epidemics were collected. These epidemics represent distinct outbreak patterns and viruses, and allow evaluation of the extent to which predictions hold across epidemics. The epidemics of interest are as follows:

- **DENV-1:** 601 dengue 1 virus isolates sampled between 2003 and 2008 in in South East Asia. Dengue fever is caused by this virus, which is transmitted by mosquitoes most common in tropical areas [34].
- **SARS:** 73 Severe acute respiratory syndrome cronavirus (SARS-CoV) sequences collected during an outbreak of the virus in Southern China between November 2002 and July 2003 [35].
- **H3N2:** 448 sequences from a 2003-2004 H3N2 seasonal influenza A epidemic [36].

5.2.2 Run data

Few sources exists from where $(BEAST, LogFile)$ pairs, which fully describe an analysis and its results, can be collected for the purpose of training prediction algorithms. Hence, the procedure outlined in Algorithm 1 was implemented in Python to generate training data. $getRandomTimeSpan$ finds a random start and end date during the outbreak, $getRandomSequences$ selects a random number of sequences that were sampled between these dates (selecting at least 30 sequences, and no more than 300 due to prohibitive runtimes), and $uploadAndRun$ uploads the BEAST xml file to the Kimura server and starts the analysis, recording the CPU time for each run with the unix command $time$. On the completion of each run, $(BEAST, LogFile)$ pairs were downloaded from Kimura and added to the training data set.

Algorithm 1 Training data generation

$numRuns \leftarrow$ number of datapoints to generate
$sequences \leftarrow$ hash mapping sequence ids to nucleotide sequence
$dates \leftarrow$ list of decimal dates of sampling for every sequence
$numRun \leftarrow 0$
while $numRun < numRuns$ **do**
  $timespan \leftarrow$ GETRANDOMTIMESPAN$(dates)$
  $sequenceSelection \leftarrow$ GETRANDOMSEQUENCES$(timespan, sequences)$
  $runFile \leftarrow$ GENERATEBEASTFILE$(numRun, sequenceSelection)$
  UPLOADANDRUN$(runFile)$
  $numRun \leftarrow numRun + 1$
end while

5.3 Learning a prediction rule

A final training set of 182 $(BEASTXMLFile, LogFile)$ pairs was arrived at - 97 from H1N12009, 46 from DENV-1, 19 from H3N2, and 20 from SARS. While this set is small, time is limited and just these runs took a total of 1626 CPU hours to complete.

5.3.1 Feature extraction

The log file of each training item was parsed for CPU hour runtime of each analysis. The sum of the effective samples sizes (for the three parameters of interest) was computed using $LogAnalyser$, and this sum divided by the run time to give the ESS per hour for the run. The number of sequences and unique sites were also extracted from the BEAST XML File. Each data item was then transformed into the following feature representation:

```
(runTime,essSum,numSeqs,uniqueSites,numSeqs*uniqueSites,essPerHour)
```

The number of unique sites is defined as the number of unique columns in the sequence alignment. A very diverse set of sequences will have more unique sites

than a less diverse set. Non-unique columns are collapsed during the MCMC, reducing the number of computations per step of the chain. More specifically, the product of the number of sequences and unique sites defines the dimensions of a matrix that is used to carry out computations during every step of the chain. For brevity, the quantity representing the number of sequences $\times$ unique sites in a dataset will be referred to as the dimension of the data, or 'data dimension', and can be viewed as a measure of information content of the dataset.

5.3.2 Results

Predicting ESS per hour

The ESS per hour is important as it gives an idea of how long a set of sequences with a certain data dimension will take to give good approximations of evolutionary parameters. Figure 5.2 plots the data dimension verses the ESS per hour for all training data. This plot shows that a clear relationship between data dimension and ESS per hour was found, with the relationship generally holding across sequence datasets from distinct viral pathogens. Pearson correlation coefficients between ESS per hour and data dimension ranged from 0.844 to 0.898 on the four datasets.

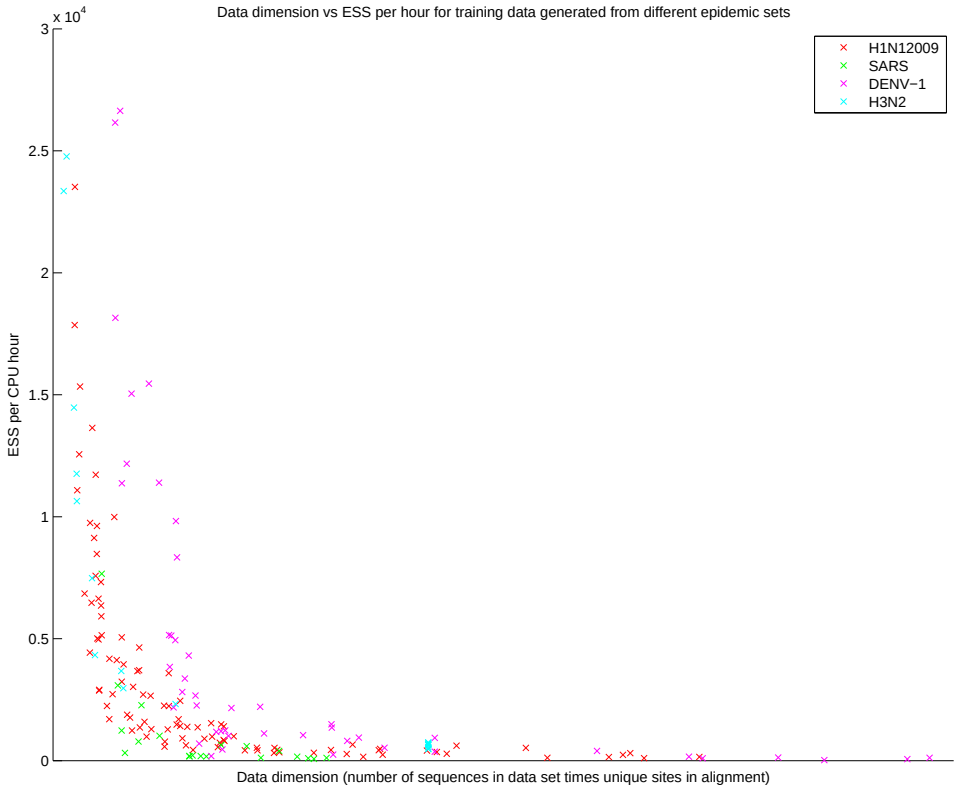


Figure 5.2: ESS per hour versus data dimension

Figure 5.3 shows an exponential curve that was fit to the H1N1 training data using

the Matlab curve fitting tool $cftool$, which gives the regression function $ESS/hour = 3.502e + 04 \times exp(-0.0002126 \times (dataDimension)$. This curve fits the data with a Root Mean Squared Error (RMSE) of 1905. Figure 5.4 shows the same plot with both axes on a log scale.

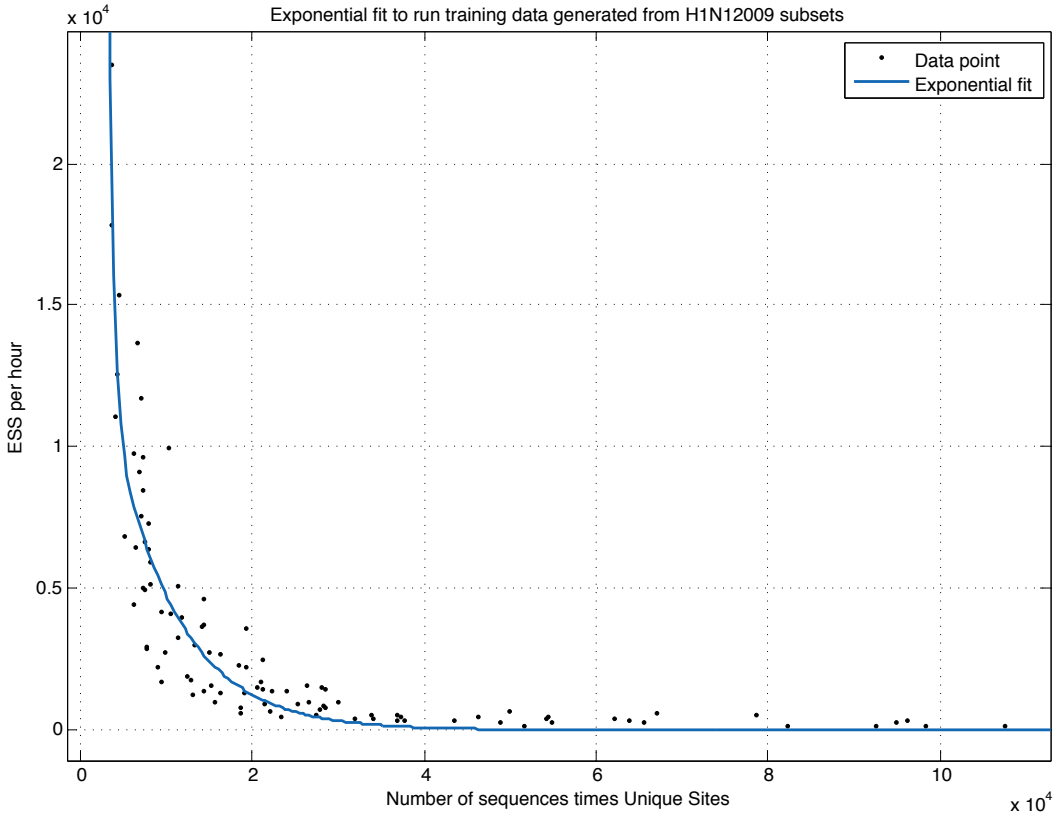


Figure 5.3: Exponential curve fit to training data

The RMSE of the fit of non-H1N12009 run datasets to the H1N12009 trained regression curve was computed, and is shown in Table 5.1. The SARS dataset gave the lowest RMSE error, as this dataset is much smaller than the other viruses (73 sequences) and even randomly selected subsets give a cluster of training points seen in Figure 5.2 that lie close to the curve, explaining the low RMSE. It is surprising that H3N2 has a worse fit than DENV-1 as the H3N2 pathogen is more related to H1N1, both being Influenza A viruses.

Dataset	**RMSE**
SARS	1.19×10^3
DENV-1	9.35×10^3
H3N2	8.87×10^4

Table 5.1: Goodness of fit of other non-H1N12009 data

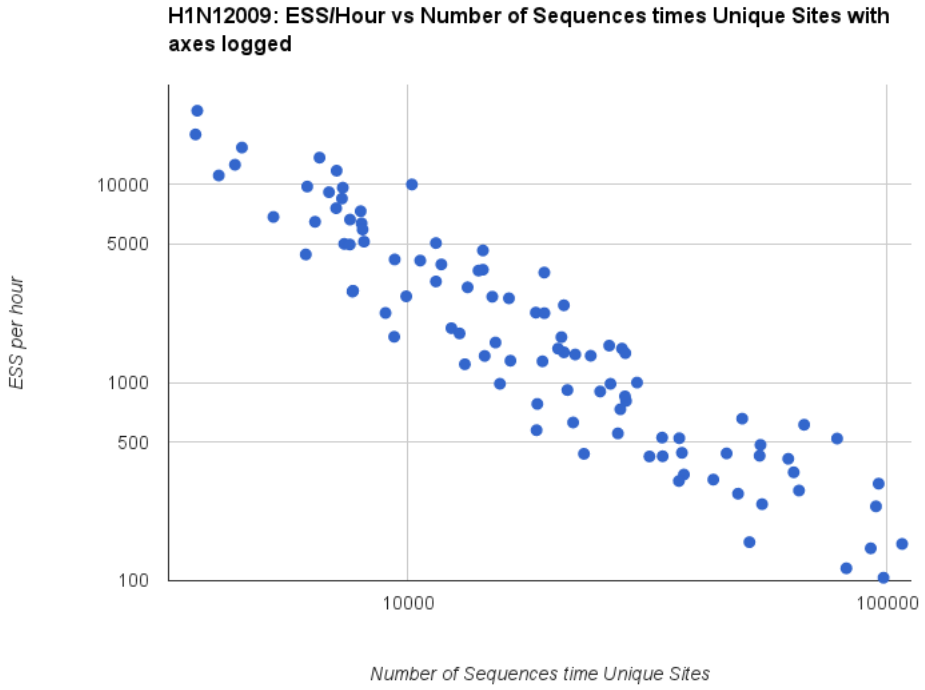


Figure 5.4: ESS per hour versus data dimension on a log scale

Other predictions

The relationship between data dimension and ESS per sample, samples per hour, and run time were also computed and curves fit to each. For example, a linear relationship was found between run time and data dimension with the fit $runTime = 0.0002003 \times (dataDimension) + 2.335$. This line gives a RMSE of 1.35 hours on the training data. However, these results are with a fixed chain length of 100 million steps, and the ESS per hour is of more interest. A graph of the runtime fit is shown below. Another experiment was carried out to see how varying the chain length affected ESS/hour, and a weak positive relationship was found reflecting the fact that longer chains will allow better mixing, allowing ESS to accrue at a faster rate over time.

5.4 Discussion

These results show that data dimension is predictive of many features of BEAST analysis. While these predictions will need to be adjusted for different evolutionary models (especially the relaxed clock assumption), these results show that data seen in past epidemics give reasonable priors on the runtime and ESS per hour for other types of epidemic data.

The ESS per hour is an important measure as an estimate with a higher ESS is a better approximation of a certain evolutionary parameter. Hence, a user may

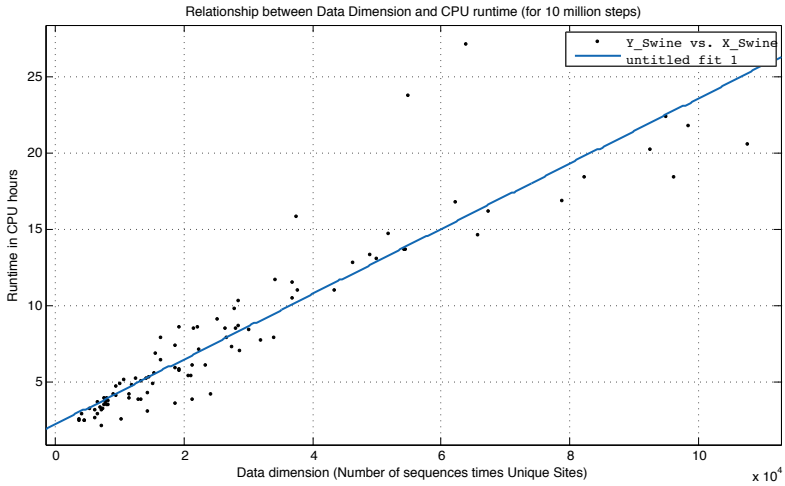


Figure 5.5: Data dimension versus CPU runtime of BEAST run

specify the minimum ESS that all runs need to achieve. On the arrival of new data, WILDEBEAST extracts the data dimension of the cumulative dataset, and predicts the required chain length and runtime for an analysis on this dataset to achieve this ESS. If this is found to violate run time restrictions that a user may have (for example, all runs must complete within a week as this is the interval in which formal estimate reports are reported to the WHO), WILDEBEAST can subsample the cumulative dataset to a set with dimension that will achieve this ESS in the runtime required.

Six regression curves, trained from this dataset, were implemented into the $corejobs$ package of WILDEBEAST. Some of the fits presented above were adjusted to reduce over fitting to the training data (by fitting simpler exponential models in Matlab), as very large or small data dimension were found to give extreme predictions. Predicted run times and ESS are displayed on the Runs information and Decisions pages. The system can easily be extended to store the predicted and true outcome of runs, gathering more training data over time to learn better prediction rules.

Chapter 6

Sequence selection

This chapter discusses approaches to the problem of sequence selection, with Section 6.1 setting forth the motivations for the problem. Measures for the quality of subsets are introduced in Section 6.2, followed by a description of the implemented sequence selection algorithms in Section 6.3. Finally, an evaluation of each algorithm is presented in Section 6.4.

6.1 Motivation

Results in the previous chapter show that ESS per hour decreases exponentially with data dimension, of which number of sequences is a major component. While the previous prediction rules can be used to estimate the number n of sequences for which an analysis is predicted to satisfy runtime or ESS constraints, this chapter explores methods for actually selecting n sequences from the total cumulative dataset N. The explosion in sequence dataset sizes is evident in the H1N12009 data, and extending ad hoc methods used for filtering in published epidemic studies to more general techniques is a crucial preparatory step for future outbreaks, yet no study has been carried out to explore robust methods for sequence selection.

6.2 Measures

In this section, measures defined on n sized subsets of sequence data and are introduced and linked to the quality of a subset.

6.2.1 Spread and Entropy

Ultimately, BEAST attempts to fit a number of hidden parameters of an evolutionary model that defines the process by which all observed genomic sequences came about. It follows that a set of n sequences that represent the progression of a viral pathogen over a longer timespan is desirable, as the sequences provide clearer information about hidden parameters, such as the rate of evolution. The spread of a set of n sequences is defined as the decimal year difference between oldest

sampled sequence and the most recent, converted to days.

Another measure that attempts to model the distribution of sequences over days is entropy, defined as:

$$-\sum_{i=1}^{d} p(x_i) \log p(x_i) \tag{6.1}$$

where d is the span of days of all sequences, and $p(x_i)$ is the probability that sequence was sampled on this day, estimated from maximum likelihood counts of dates from the n sequences.. Entropy is maximised with uniform distributions, i.e. sets of n where every sequence was sampled from a different day. It follows that a selection method should aim to maximise entropy.

6.2.2 Distance based measures

The location of where an infected human from which an isolate sampled can be parsed into GPS coordinates, and these coordinates can be clustered to define epidemic clusters, from which sequences can be selected in a way that maximises diversity with respect to location. An extension to this would be try to maximise the diversity of sequences in the subset we select with respect to location, date, and nucleotide sequence. Diversity can be measured with respect to these features by representing each sequence in vector space and selecting subsets that have maximal distance in this space (after normalising each dimension).

6.3 Sequence selection algorithms

This section describes the implementation of several subset selection algorithms that use the measures introduced above. In general, only subsets where $N \geq 30$ are considered as BEAST estimates run on fewer than 30 sequences are not alwaus meaningful. The subset of size n is denoted s.

6.3.1 $maxSpread$ algorithm

The $maxSpread$ algorithm was implemented. This algorithm selects s by randomly selecting subsets of size n until a set is found that has a spread that exceeds $spread_{thres}$, and such that fewer than $repeats_{thres}$ sequences are sampled from the the same day as one another. $spread_{thres}$ was set to the spread of N, hence this algorithm always selects at least the oldest and most recent sequences. If is not possible to find a subset with fewer than $repeats_{thres}$ repeats, this threshold is adjusted. While a naive and randomised algorithm, $MaxSpread$ can be performed relatively quickly on large datasets, especially when location information is unavailable.

6.3.2 Cluster selection

The cluster selection algorithm was implemented in Python using functions implemented in the SciPy, GeoPython, and BaseMap libraries. This algorithm first parses sequence identifiers to extract words representing the location where a sequence was sampled from. The Google Geocoding API is then used to carry out geocoding from these words to retrieve a GPS coordinate. For example, the following identifier

`A/NewYork/2009|NorthAmerica/USA/NewYorkstate/SuffolkCounty|2009.321`

is parsed to 'Suffolk County in New York' through string manipulations, which when queried through the Geocoding API returns the correct coordinate of $(40.98, -72.61)$, disambiguating this location correctly from Suffolk County in the UK. Daily query limits are applied by the Gecoding API, hence an archive of sequence ID to coordinate mappings was created to reside within the file system of an epidemic, reducing the number of queries.

Sequences are then represented by their coordinates, and the geodesic distance in miles between two sequences computed using functions implemented in the GeoPython library. Hierarchical clustering on these distances is then carried out using implementations provided by SciPy. Figure 6.1 shows an example of the clusters selected by the algorithm, where size and colour of points on the map are proportional to the number of sequences in a cluster. Hierarchical clustering allows the prior specification of the number of clusters, and this can be set to n, after which s can be generated by selecting 1 sequence randomly from each cluster. If fewer than n clusters are generated, sequences are picked iteratively from each cluster until n is met.

6.3.3 $vectorDist$ algorithm

The $vectorDist$ method extends the above distance-based selection method to define an algorithm that uses three features of each sequence - location, date, and nucleotide sequence - to project the set of N sequences into a new distance space. From this space, the algorithm seeks to pick n sequences that have maximal pairwise distance in this space. Algorithm 2 outlines this procedure and was implemented in Python.

The weighting of each feature of a sequence in the computation of pairwise distances needs to be considered. While hamming distance could be used to count the number of sites in which two sequences differ, it is not clear how to factor in this distance equally with location and date differences when defining the total distance between two data points. A method that uses such an approach would seek to maximise nucleotide diversity as much as spread and location. This is not a sound criteria for choosing sequences that represent the evolution of a pathogen, as observing a sequence at the start of an epidemic and another with the same or similar genome much later reveals much about the evolutionary process, and the inclusion of such sequence pairs in a subset would be valuable.

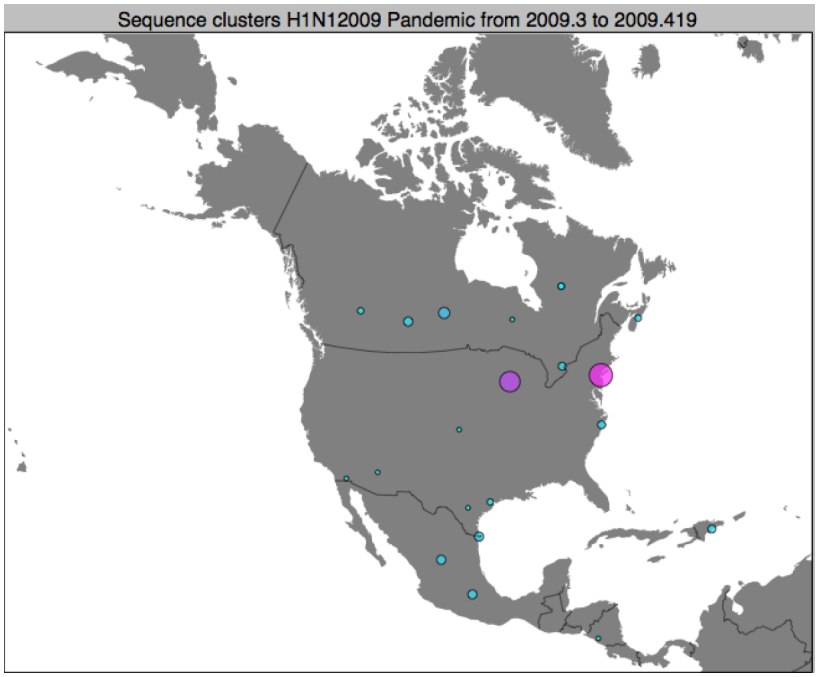


Figure 6.1: Clusters selected using the hierarchical clustering algorithm using all sequence data at the end of week 9 of the H1N12009 pandemic. Map generated using the Python library Basemap.

As a result, a method for weighting each feature in the distance computations was introduced through the $nucleotide_{weight}$, $location_{weight}$ and $date_{weight}$ parameters of the algorithm. For brevity, $vectorDist(1, 2, 3)$ means vectorDist($date_{weight}$ = 1, $nucleotide_{weight}$=2, $location_{weight}$=3). These weighting are implemented in WILDE-BEAST to allow a user to specify weights, allowing for flexibility and the experience of evolutionary epidemiologists to inform the selection process.

Algorithm 2 $VectorDist$ algorithm

$sequences \leftarrow$ hash mapping sequence ids to nucleotide sequence
$sequenceLocations \leftarrow$ COMPUTELOCATIONS($sequences$)
$sequenceDates \leftarrow$ COMPUTEDATES($sequences$)
$featureVectors \leftarrow$ VECTORISE($sequences, sequenceDates, sequenceLocations$)
$distMatrix \leftarrow$ COMPUTEDISTANCES($featureVectors$)
$n \leftarrow$ size of subset
$s \leftarrow$ set of selected sequences so far
while $sizeofs \leq n$ **do**
  $sequenceSelection \leftarrow$ GETMOSTDISTANTPAIR($distMatrix$)
  $s \leftarrow s \cup sequenceSelection$
end while

6.4 Evaluation

Published reports on the H1N12009 pandemic estimate the mean evolutionary rate to be 3.66×10^{-3} substitutions/site/year and the time of the most recent common ancestor (TMRCA) of samples to be 21 Jan 2009 [16]. Selection algorithms were used to generate subsets s of size n from the cumulative data N available at each the end of each week of H1N12009, starting from week 8 to week 14. Week 8 was chosen since N at this point is large enough to reflect a situation where subsampling may realistically need to be applied. BEAST runs with 100 million chain steps was carried out on each s to ensure convergence, and the absolute distance of the posterior mean estimates from the published values plotted for each method.

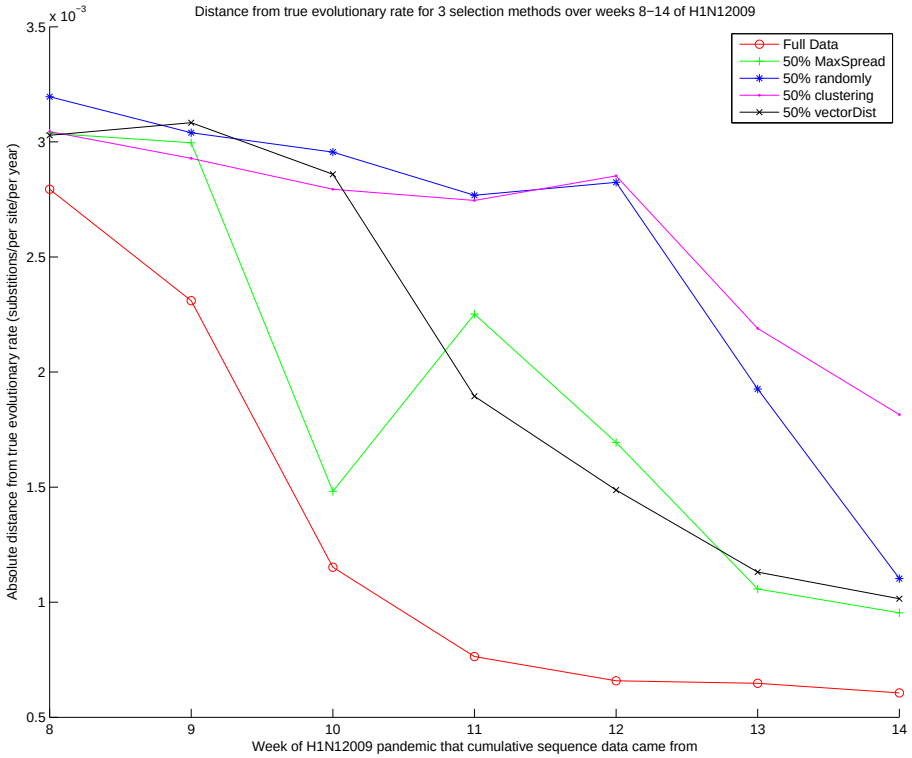
6.4.1 Evaluating three selection methods

Figure 6.2a and b shows plots for the evolutionary rate and TMRCA estimates, with $n = 0.5N$, using three subset selection methods - cluster selection , $maxSpread$, and $vectorDist(1, 1, 1)$. A baseline method of selecting subsets with minimal spread and ceiling of using the full data are also plotted. Results from $maxSpread$ are averaged over two runs per week.

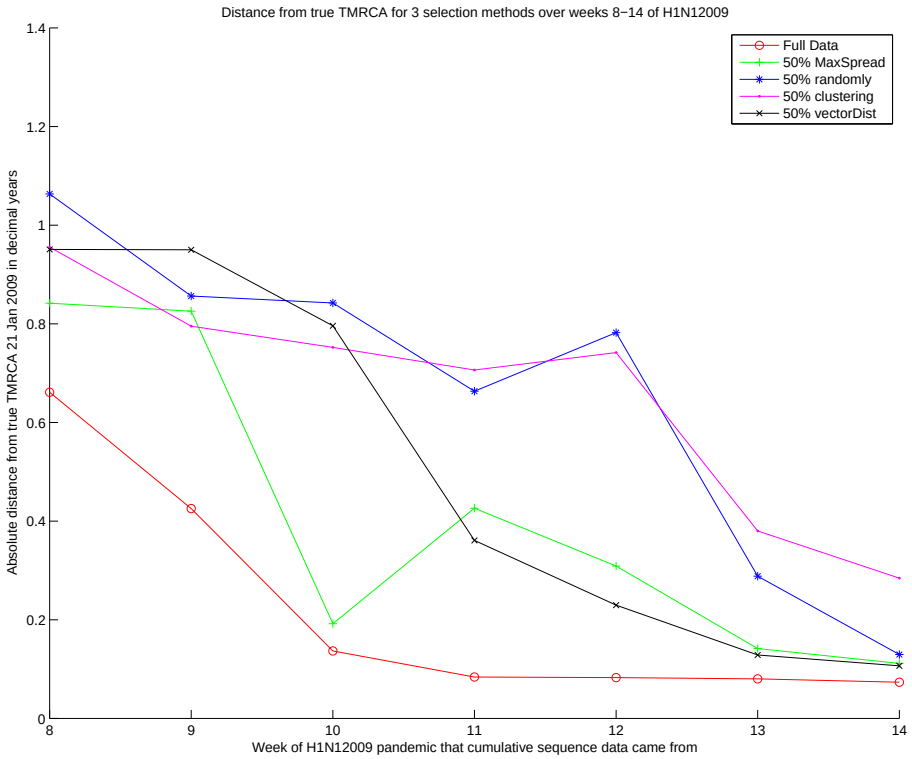
In Figure 6.2a, the $maxSpread$ and $vectorDist$ methods perform the best - with each giving the closest estimates to the full data method for 3 out of 7 of the weeks and the clustering method performing best once. These results give evidence that a larger spread in a set of sequences give better evolutionary estimates through BEAST. $maxSpread$ results show a large increase in accuracy from week 9 to 10 , before losing accuracy sharp from week 10 to 11, and this reflects the randomised nature of this algorithm. $vectorDist$ stands in contrast to this, as the accuracy of estimates given through this method more closely follow the trend set by using the full dataset. These results suggest that $vectorDist$ is a more robust selection method that better reflects datasets of size N even when using only half the sequences. Cluster selection performed poorly, as it does not take spread or date information into account when making selections nor the relative size of each cluster into account when making selections. TMRCA estimates in Figure 6.3b show very similar trends, showing that results hold across different evolutionary parameters.

6.4.2 Evaluating $vectorDist$ weightings

Different weights of the sequence features for the $vectorDist$ algorithm were investigated by carrying out the same evaluation as above comparing $vectorDist(1, 1, 1)$, $vectorDist(2, 1, 1)$, $vectorDist(1, 0.75, 0.75)$ and $vectorDist(1, 0, 0)$. The results for this test are shown in Figure 6.3. These results show $vectorDist(1, 1, 1)$ consistently performs better than all methods other than $vectorDist(0, 1, 0)$, again following the trend set by the full dataset. $vectorDist(0, 1, 0)$ performs very closely to the full dataset until week 11, before giving the worst results from week 11 onwards.



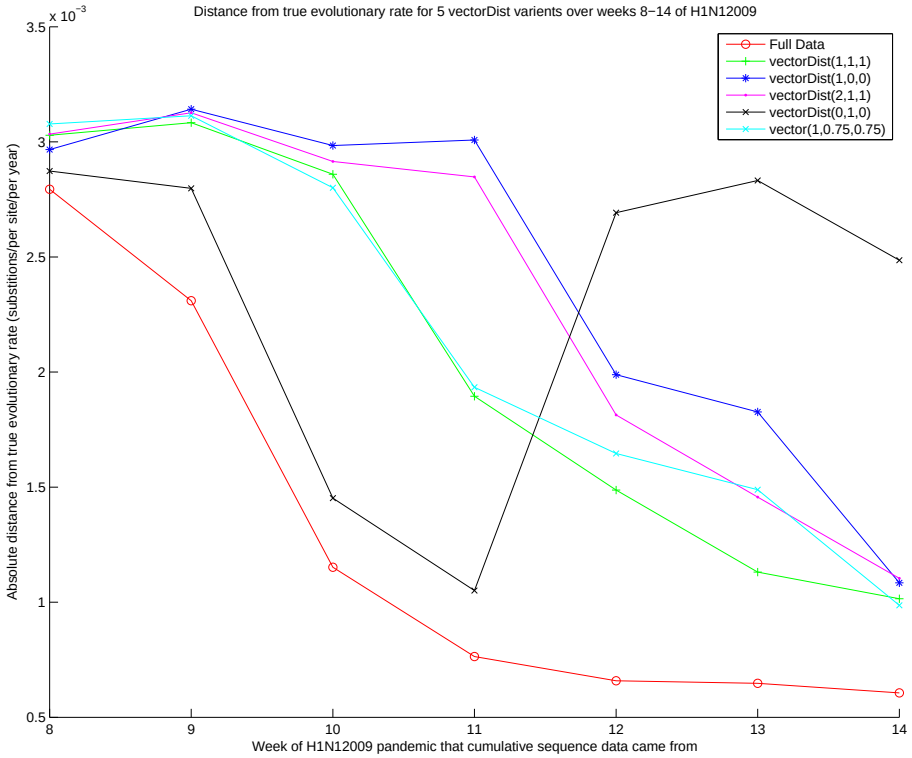
(a) Estimates for evolutionary rate using five different subset selection methods



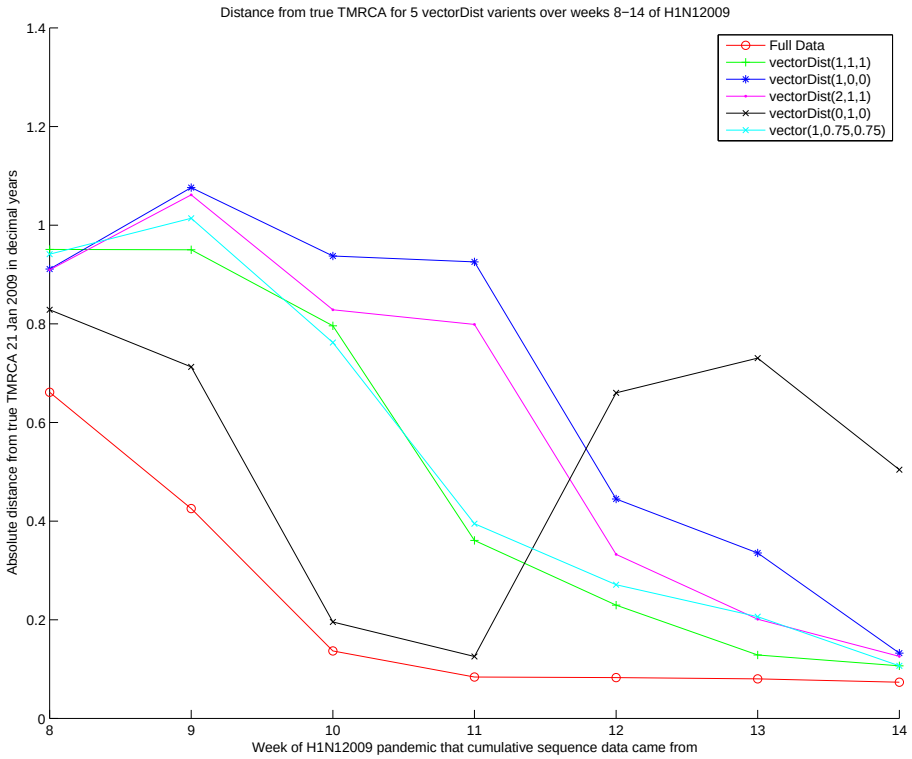
(b) Estimates for TMRCA using five different subset selection methods

Figure 6.2: Selection method comparisons

This is evidence for the previously discussed conceptual problem of only maximising sequence diversity. Overall, run times for all analysis using downsampled sets s were less than half the runtime of full datasets.



(a) Estimates for evolutionary rate using five vectorDist weightings



(b) Estimates for TMRCA using five vectorDist weightings

Figure 6.3: VectorDist selection comparisons

Chapter 7

Decision Making

The decision component of WILDEBEAST defines the autonomous behaviour of the system, and how the methods introduced above function together. Three major tasks are carried out by this component:

1. Monitoring, logging, and summarising BEAST analyses and cumulative sequence knowledge.
2. Deciding how and if to start new BEAST analysis on the arrival of new sequence data.
3. Deciding which BEAST analysis to report parameter estimates from.

Section 7.2, 7.3 , and 7.4 explore how WILDEBEAST carries each of these tasks. A number of user-set operational settings guides the systems decision making processes, and these are introduced below. The learning and sequence selection components, discussed previously, and a novel sequence insertion method for reducing burn-in of run are also key components in completing the above tasks.

7.1 Global controller process

The World Health Organisation emphasises the importance of global surveillance during epidemics, during which systematic data collection and analysis are critical [21]. WILDEBEAST carries out these functions, albeit operating only on a specialised subset of epidemiological data - genomic sequences of a viruses. A global controller process, an outline of which is given in Algorithm 3, was implemented to run at intervals specified by the user-set parameter $reportInterval$ (set to a default of 5 minutes). This process checks for new sequence data, initiates the run management and parameter estimate decision processes, and generates summaries to present the state of cumulative sequence knowledge through web displays.

Section 7.3 and 7.4 describe how the processes represented by the $makeRunDecisions$ and $getBestEstimate$ operate. The $updateAllRuns$ function uses Run objects (see Chapter 4) to update the summary files and phylogeny for every analysis, as well as checking for completed runs and updating their status in the databases.

Algorithm 3 WILDEBEAST controller

repeat
  $epidemics \leftarrow$ a list of all currently tracked epidemics
  $time \leftarrow$ current server time
  $logs \leftarrow \emptyset$ stores all logs information for this timestep
  for all $epidemicID \in epidemics$ **do**
    $newSequences \leftarrow$ CHECKFORNEW$(epidemicID)$
    if $|newSequences| \geq 1$ **then**
      $actions \leftarrow$ MAKERUNDECISIONS$(epidemicID)$
    end if
    $errors \leftarrow$ UPDATEALLRUNS$(epidemicID)$
    $estimates \leftarrow$ GETBESTESTIMATE$(epidemicID)$
    UPDATEWEBDISPLAYS$(epidemicID)$
    $logs \leftarrow logs \cup estimates \cup errors \cup actions$
  end for
  LOGTHISTIMESTEP$(logs, time)$
  SLEEP$(reportInterval)$
until WILDEBEAST service is stopped

Three log files are maintained: one for actions taken by the run management system, a second for logging errors that occur during the run update process (due to failed runs or other technical issues), and a third for storing parameter estimates for each run for each epidemic. The $updateWebDisplays$ updates the Data, Decisions, and Summary pages of each epidemic to reflect actions taken, lists the new cumulative sequence data and its parameters, and generates a map of all sequences if locations are available.

7.2 Run management

7.2.1 Operational settings

The run management module is parametrised by seven user-specified settings - each described in detail in Table 7.1. These parameters allow a significant level of user input into how this module operates, and can be updated at any time during an epidemic. It is intended that these parameters are set by expert evolutionary epidemiologists as an epidemic progresses, enabling fine tuning of how the system operates autonomously.

The $runtime_{limit}$ setting is motivated by the fact that epidemic monitoring sometimes requires reports at set intervals - for example the WHO have used weekly intervals for reporting in the past [21]. $ESS_{threshold}$ follows from the discussion set forth in Chapter 2 - the system should aim to report estimates that are good approximations for parameters of interest, and hence should start runs in a way that aims to satisfy some minimum ESS requirement. $concurrent_{limit}$ allows computational resource limits to be defined - for example, a server may only have enough

cores to effectively sustain 8 concurrent BEAST runs.

Setting	Type	Description
$auto$	Boolean	Specifies whether WILDEBEAST should try to start runs autonomously. If false, will only make recommendations on the 'Decisions' webpage.
$concurrent_{limit}$	Integer	Sets the limit on the number of concurrent analyses. Allows deployment of WILDEBEAST in resource constrained settings. Set to -1 if there is no limit.
$runtime_{limit}$	Double	Specifies, in hours, the upper limit on the runtime of all analyses. Sequence selection is informed by this limit. Set to -1 if there is no limit.
$ESS_{threshold}$	Double	The minimum ESS that all analyses run by the system should achieve for the three evolutionary parameters of interest. Sequence selection is informed by this limit. Set to -1 if there is no limit.
$percentDown$	Double	Specifies a simpler downsampling scheme in the absense of runtime or minESS limits.
$selection$	String	Specifies the sequence selection method to be used for downsampling. Options: $maxSpread$, $vectorDist(x, y, z)$, $clusterSelect$ or $random$.
$insertion$	Boolean	Sets whether the system should attempt to use the sequence insertion method when starting new runs.

Table 7.1: The operational parameters for run management in WILDEBEAST

7.2.2 Run creation

On the detection of new data, and if $auto$ is set to $True$, the system will attempt to start a new BEAST analysis following the pseudo code presented in Algorithm 4. The data dimension of the new cumulative dataset is extracted, and used as input to the learnt prediction algorithms (see Chapter 3) to predict the chainlength of a run that will achieve $minESS$ in the parameters of interest. The expected runtime of this analysis is then predicted, and should it exceed $runtime_{limit}$ hours, the system will attempt to downsample the sequences with the method specified in $selection$ to a number of sequences that is closest to the runtime limit but achieves or exceeds the ESS threshold. If the runtime limit and ESS thresholds are not set, WILDEBEAST will simply start a run on all of the cumulative sequence data.

A naive operating mode was also implemented that is identical to Algorithm 4, except that $selectSequences$ will always downsample the cumulative dataset to $percentDown$ of the total size when starting a new run, while ensuring that the chainlength is long enough for $ESS_{threshold}$ is met, but not adhering the runtime limit. This mode was introduced as always trying to meet both the ESS and runtime threshold will result in the system reaching a sequence limit on all analysis,

and this may interfere with parameter estimates in the long run.

Algorithm 4 Run generation

$sequences \leftarrow$ hash of previous sequence knowledge, including new data
$phase \leftarrow$ current epidemic phase (see Section 7.3.1)
$activeRuns \leftarrow$ number of currently running analyses
if $ESS_{threshold} \neq -1 \wedge runtime_{limit} \neq -1$ **then**
  $dataDimension \leftarrow$ EXTRACTFEATURES$(sequences)$
  $predictedRuntime, chainLength \leftarrow$ PREDICTRUNTIME$(dataDimension, ESS_{threshold})$
  if $predictedRuntime \leq runtime_{limit}$ **then**
    $newRunID \leftarrow$ STARTRUN$(sequences, chainLength)$
    $phase \leftarrow 1$
  else
    $seqRunset, chainLength \leftarrow$ SELECTSEQUENCES$(sequences, selection)$
    if $activeRuns < concurrent_{limit}$ **then**
      $newRunID \leftarrow$ STARTRUN$(seqRunset, chainLength)$
      $phase \leftarrow 2$
    else
      $stopRunID \leftarrow$ STOPRUN()
      $newRunID \leftarrow$ STARTRUN$(seqRunset, chainLength)$
      $phase \leftarrow 3$
    end if
  end if
else
  $newRunID \leftarrow$ STARTRUN$(sequences)$
end if

7.2.3 Phasing

Since 1999, the WHO has utilised six pandemic phases to guide decision making and actions on a global scale [22]. This inspired a simple phasing system for WILDEBEAST which classifies epidemics based on their cumulative sequence data sizes. Should users have strict runtime, ESS, and computational resource limitations, the phasing system notifies them of when WILDEBEAST starts discarding data or stopping analyses in order to meet these requirements, which may prompt a user to adjust their settings. The phasing system also helps capture the wide variety in sequence availability for different epidemics. The phases implemented are described in Table 7.2.

The decision making system could be extended in a number of ways that rely on the phasing system - for example, once an epidemic reaches phase two due the arrival of new data, WILDEBEAST could automatically notify evolutionary epidemiologists using the system by email and suggest allocation of more computational resources for BEAST analysis for that epidemic. WIDLEBEAST could also suggest a new subset selection method or weighting for $vectorDist$ that helps filter data more effectively. More phase-specific strategies can be implemented in future work, such

Phase	Condition	Description
One	Predicted runtime on cumulative data to achieve $ESS_{threshold}$ is $\leq$ $runtime_{limit}$	Characterises epidemics with manageable, even scarce data, example: The Middle East respiratory syndrome coronavirus (MERS-CoV).
Two	Predicted runtime on cumulative data to achieve $ESS_{threshold}$ is $>$ $runtime_{limit}$	Epidemics with an abundance of data, but few active analyses.
Three	Predicted runtime on cumulative data to achieve $ESS_{threshold}$ is $>$ $runtime_{limit}$ and $activeRuns$ = $concurrent_{limit}$	Epidemics with an abundance of data, requiring downsampling, and scarcity of computational resources.

Table 7.2: WILDEBEAST phase descriptions

as switching evolutionary models based on the available data, which has been shown to be important in previous work [18].

Phasing information is displayed on the Decisions page, which also allows users to specify operational parameters discussed above. An example of a decision page is shown in Chapter 3 (Figure 3.4b).

7.2.4 Sequence addition

A central issue with starting a BEAST analysis on the arrival of new data is that the chain defined by this new MCMC must undergo a burn-in period during which it attempts to find a region of high density in the posterior distribution. Parameter samples from this period are unreliable. When simply starting a new BEAST analysis from scratch on the arrival of new data, results from previous runs, especially those that have converged, are not taken into account, resulting in a unnecessary delay in a systems ability to get timely and reliable estimates of parameters of interests at a time where updated estimates are needed.

The sequence addition method aims to overcome this problem by copying over the final values sampled for evolutionary parameters from a previous run that has converged, and makes use of these values as starting values of the new run. A challenge exists when transferring the phylogeny estimated from a previous run as a starting tree in the new run. The most significant part of the probabilistic model in affecting burn-in and likelihood is the phylogeny, as the space of possible topologies is intractable and the distribution over these topologies is peaked at an extremely small region of the space. New sequences must thus be inserted by some method into the previously estimated phylogeny.

Algorithm 5 outlines the proposed approach for this task. An overview of this process is firstly to decide which previous run to use results from, then compute the MCC tree of this run (which summarises the phylogeny estimated so far - see Sec-

Algorithm 5 Sequence insertion

$recentID \leftarrow$ FINDMOSTRECENTRUN$(epidemicID)$
$oldSequences \leftarrow$ GETSEQUENCES$(recentID)$
$newSequences \leftarrow$ new sequences that have arrived
$distMatrix \leftarrow$ GETDISTANCES$(newSequences, oldSequences)$
$evolutionaryRateEstimate \leftarrow$ GETRATE$(epidemicID)$
$treeMCC \leftarrow$ GETMCC$(epidemicID)$
$minHeight \leftarrow \epsilon$
for all $sequence \in newSequences$ **do**
  $closestSeq \leftarrow$ FINDCLOSESTSEQUENCE$(distMatrix)$
  $distance \leftarrow$ GETDISTANCE$(sequence, closestSeq)$
  $timeClosest \leftarrow$ GETDATE$(sequence)$
  $timeNew \leftarrow$ GETDATE$(closestSeq)$
  $timeForDist \leftarrow distance/evolutionaryRateEstimate$
  $estimateInsertHeight \leftarrow (timeForDist - |timeClosest - timeNew|)/2$
  $insertHeight \leftarrow$ MAX$((estimateInsertHeight, minHeight)$
  $treeMCC \leftarrow$ INSERTSEQUENCETOTREE$(sequence, insertHeight)$
end for
$startingVals \leftarrow$ GETLASTSTEPPARAMVALES$(recentID)$
$beastFile \leftarrow$ GENRUNFILE$(startingVals, MCC)$
STARTRUN$(beastFile)$

tion 4.2.1). Then for each of the new sequences, insert each at a location in the phylogeny near the sequence that has the most genetic similarity to it. This genetic similarity is computed using hamming distance between the sequences divided by the sequence length (number of sites). Dividing this quantity by the current estimate of the evolutionary rate (extracted from the previous run, or as reported by the system), gives a rough estimate of the time between the new sequence and its most genetically similar partner on the tree. This is then transformed to find the height at which a common ancestor node for the two sequences should be inserted in the tree.

The Java Evolutionary Biology Library (JEBL) was used to parse MCC trees from newick format (a text based format for specifying trees), and tree objects used to carry out the insertion of sequences. The values of evolutionary parameters on the last recorded step of the previous run are also extracted. The newick of the new MCC tree and values of parameters are specified as priors to generate a BEAST run file, and the new run in started. Chapter 8 shows that this method performs considerably better than simply starting a new run in which the starting tree is randomly generated.

7.3 Reporting

7.3.1 Modelling quality of runs

Algorithm 4 makes use of a $StopRun$ function call to choose a run to stop to allow another to be started, should the active run limit be reached. In addition, the system must decide which run to report parameter estimates from. To carry out these tasks, metrics were introduced that allow WILDEBEAST to reason about the quality of runs. These inform decisions that need to made in situations such as when a substantial set of new sequence data arrives just after the system started a new run on an old dataset. The desired action in this situation is to stop the previous started run and start a new run in its place that takes this new data into account.

Two measures were proposed for each run - the first to model the potential of a run (defined by a static measure of how well it represents evolutionary knowledge of a pathogen), and the second a dynamic measure of the realised potential of a run, based on how good an approximation the posterior mean estimates given by the run are at a specific point in time. At each time interval, the realised potential for each run is computed and runs and sorted by this measure. The run with the smallest realised potential is stopped, and parameter estimates from the run with the highest realised potential are displayed on the summary page, an example of which was shown in Chapter 3 (Figure 3.2a).

To enable fast decision making and reporting, the potential of a run was defined as the spread of a dataset used for a run, and the realised potential the product of its ESS in parameters of interest so far and spread. Analysis are ordered by potential on the Runs page, and WILDEBEAST constantly weights the potential implicit in a dataset with the progress made by runs as part of its decision making processes. Other possible measures for potential could be the total pairwise distance of sequences in a run as computed by $vectorDist$, or the dimension of the dataset. More complex methods could also be used for reporting, and this is discussed in detail in Chapter 9.

7.3.2 Logging

Examples of the actions and estimates log files can be downloaded directly from WILDEBEAST. On the arrival of new data, whether a phase changed occurred is logged along with predicted runtime, ESS/hour, and what action the system took, for example:

```
2014-04-02 17:31:21 Old sequences 3. New sequences: 35.0.
    Predicted runtime: 2.58 hours (Threshold:10.0).
    Parameterss: NumSeq: 35  MinDate: 2003.13  MaxDate:
    2003.21  Spread: 29.2  Entropy: 0.422000516883  UniSites:
     175 PHASE: Remain in 1 DECISION: Did not downsample and
    started run  autoRun20140402173118 with the default chain
     length of 1000000. Sequence insertion is not enabled.
```

A line of the estimates file first gives the best reported run and its parameter estimates, and then in brackets all other runs for that epidemic and their estimates, along with cumulative run time.

```
2014-04-02 17:33:19 H1N12009: autoRun20140402173118:
  TMRCA: 6.8047109589 Rate: 1.0969E-3 Growth: 2.3376
  RealisedPot: 35834.79772 [ autoRun20140402170118: TMRCA:
  6.6995109589 Rate: 9.9313E-4 Growth: 0.6774 RealisedPot:
  0.0] (2.0166666666666666) |
```

Chapter 8

Evaluation

The previous chapter explored the decision processes of WILDEBEAST in detail, and described a logging procedure which is used to record how the system operated over a period of time. A host of parameters were also introduced. Due to time limitations and the lengthy and computationally expensive nature of BEAST analyses, it is not possible to evaluate every combination of operating modes of WILDEBEAST. Instead, evaluations were carried out to test key modes of operation. Section 8.1 sets forth the method by which WILDEBEAST was evaluated, 8.2 presents the performance of the system on H1N12009 datasets, including sequence insertion experiments. Section 8.3 finally shows evaluatations of WILDEBEAST on data from three other epidemics which each exhibit distinct data arrival patterns.

8.1 Methodology

An epidemic simulation environment was implemented by taking sequence data for each of the epidemics of interest (see Chapter 5), and dividing this data into sets representing the sequence knowledge at evenly spaced time points during each epidemic. A timing parameter was then set, specifying the intervals at which new aligned sequence data, in the form of a FASTA format file, would be uploaded to Kimura to replace the $newSeqs.fasta$ file for each epidemic. Unless otherwise mentioned, all runs were carried out with settings of the evolutionary model discussed in Chapter 4.

It is not possible to model the release of sequence data on the same time scale as it was observed in reality, as some epidemic datasets span years, hence the time scales of epidemics were adjusted. Such a scaling of time is appropriate as it reflects the fact that in future epidemics, sequence dataset sizes are expected to grow substantially due to advances in next-generation sequencing technology, and reductions in the time lag between an infection and publication of genomic sequence. Larger datasets result in longer times for chain convergence, and slower mixing, even after filtering. Saying a minute of real time is equivalent to 100 minutes of WILDEBEAST time can be seen as modelling this increase in computation.

There are no other real time viral characterisation tools that WILDEBEAST can be evaluated against, as discussed in Chapter 2. Hence evaluations focus mainly on comparing modes of operations of WILDEBEAST and their ability to converge to good estimates. The main evaluation metric was difference in estimates given by WILDEBEAST to 'gold standard' evolutionary estimates for either the growth rate, TMRCA, or evolutionary rate of each pathogen. Fold standard estimate were retrieved from previous literature. While these estimates were usually estimated with BEAST through use of a slightly different evolutionary model or dataset, these evaluations still show how well WILDEBEAST approaches estimates published for the purpose of policy making. In some evaluations, r_0 estimates are omitted because the transform to this quantity from R_0 is highly dependent on generation time assumptions of pathogens, and estimating these values is considered outside the scope of this project.

8.2 H1N12009 results

Sequence data from the H1N12009 pandemic was retrieved as both an unfiltered and filtered dataset (See Section 5.2.1). The purpose of the first two experiments were to evaluate whether the completion criteria of the project had been achieved - namely a working proof of principle using 2009 A/H1N1 pandemic influenza as exemplar. These experiments also tested if the components of WILDEBEAST were able to operate autonomously to correctly give estimates and report them over time during an epidemic. This procedure requires that the sequence selection methods, logging interface, databases, interfaces to BEAST, and WILDEBEAST controller all function smoothly together. A second goal of the evaluation was to observe how a basic decision process of subsampling effects parameter estimates in a real time setting, and how reported value differ from the true best estimates at each timestep. Gold standard estimates were retrieved to be 0.00393 sites/substitutions/year for the evolutionary rate, a growth rate of 7.2, and a TMRCA of 2 February 2009 [18].

8.2.1 Experiment 1: Filtered dataset

WILDEBEAST was deployed under two modes of operation - the first emulating a naive method of always starting a run on the arrival of new data without subsampling (Setting 1), and the second selecting 50 percent of the total sequence size at each time point using the $maxSpread$ algorithm before automatically starting a run (Setting 2). Each mode of operation was evaluated under a simulation of 8 weeks of H1N12009 , by taking the filtered H1N12009 dataset and dividing it into weekly cumulative knowledge over 8 weeks, spanning May 26 2009 to July 21st 2009. Time was scaled so that each weekly dataset was uploaded every 30 minutes - ie 30 minutes in real time represented a week during the epidemic, or a minute of real time equated to 5.6 hours of WILDEBEAST time. All runs were carried out with a chain length of 100 million steps - well over the amount needed for convergence - with no ESS or runtime threshold set, and the reporting interval was set to two minute intervals. The cumulative dataset sizes were as follows: 87, 105, 120, 135, 148, 161, 174, and 181.

True best estimates

The true best estimate at each timestep is computed by parsing the WILDEBEAST $estimates.txt$ log file to find the estimate of a parameter out of all runs that is closest to the gold standard estimate. Figure 8.1 shows these results for the evolutionary rate, plotted over 200 reporting timesteps of the evaluation run separately for both Setting and 2. Vertical lines indicate the arrival of new data on the server. The y-axis measures distance from the gold standard rate - all distances were positive, ie the rate was overestimated by all runs.

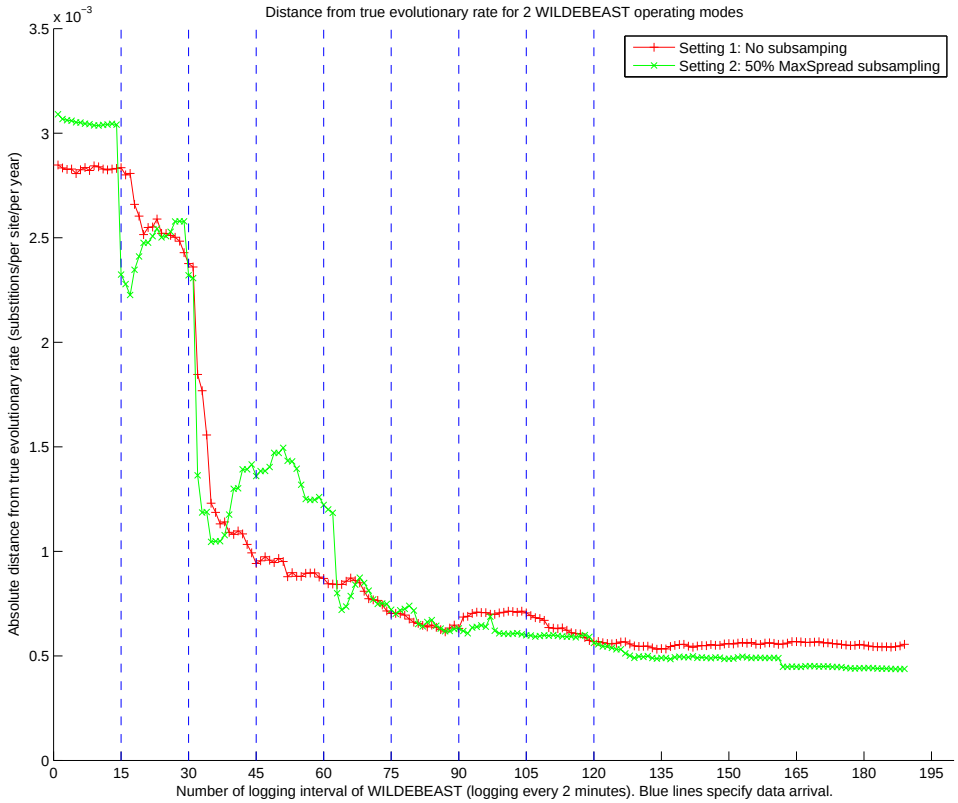


Figure 8.1: Distance of true best estimate from the gold standard evolutionary rate, under 2 modes of operation, spanning May 26 2009 to July 21st 2009 with data arriving weekly

It is clear from Figure 8.1 that WILDEBEAST functions correctly under both modes of operations. More promisingly , the general trend of estimates get much closer to the gold standard value immediately after the arrival of Week 2 and Week 3 data, and to a lesser extent on arrival of the subsequent data. While the downsampling method initially gives a worse estimate of the evolutionary rate during Week 1 (possibly due to use of the $maxSpread$ algorithm rather than $vectorDist$), the arrival of Week 2 data sees the estimates given by Setting 2 drop below those for Setting 2, before climbing and dropping again at Week 3. On inspection of which runs give the best estimates for Setting 2, it was noted that the same run gave the estimates from timestep 30 to 60, hence this decrease in accuracy shows that this run still

has yet to converge, but still gives better estimates the the runs started at the start of Week 1 or 2. From time step 60 onwards the best estimates are given by a combination of this run and the run started at timestep 60 - subsequent runs have yet to complete burn-in or sample the posterior well due to the rapid arrival of new data.

Reported best estimates

Figure 8.2 shows the same plot, expect plotting the distance from the gold standard rate the best rate reported by WILDEBEAST's decision process, which models realised potential with the product of ESS_{run} and $spread_{run}$. Estimates are not joined by a line for clarity. From these results, it is clear that the reported estimates are substantially different from the true best estimates, and that the difference between Setting 1 and 2 of operation become more substantial than in Figure 8.1. Setting 2 performs poorly until the 4th arrival of data, but gives promising results after this point.

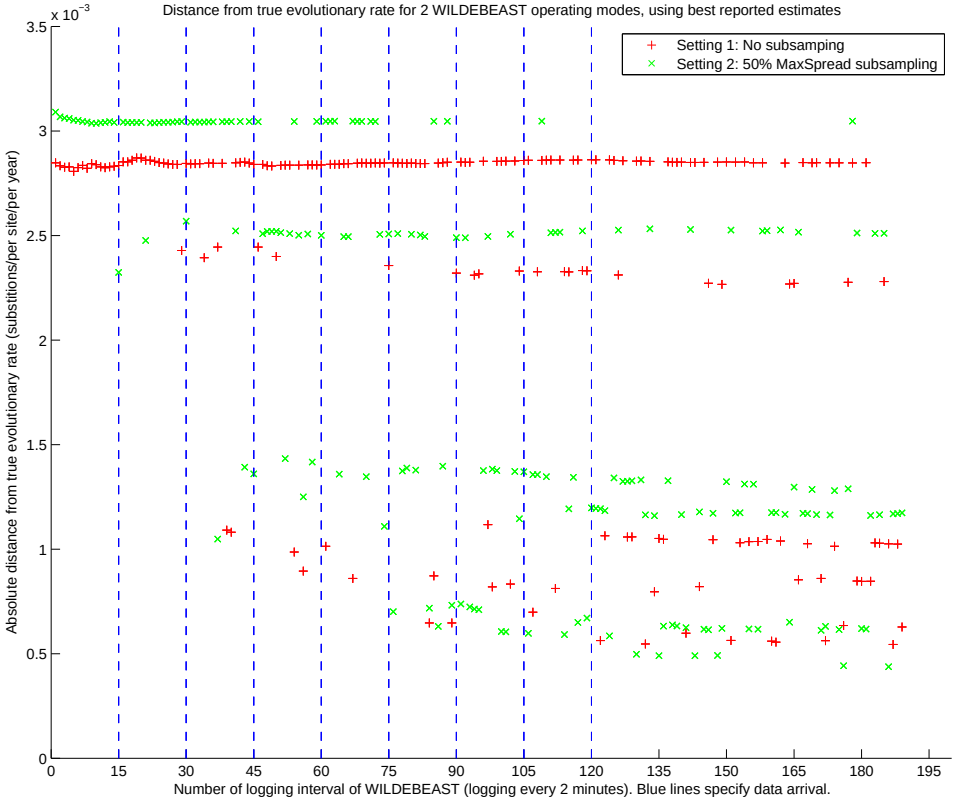


Figure 8.2: Distance of reported best estimate of the evolutionary rate, under 2 modes of operation, spanning May 26 2009 to July 21st 2009 with data arriving weekly

Figure 8.2 also shows that the proposed model of realised potential works correctly, though due to erratic jumps between estimates given from different runs (seen clearly from timestep 120 to 150), it may be better to report estimates given by the

mean of the top n ranked runs estimates, or last n best estimates, for some n, or even take a weighted sum of these based on each runs realised potential. Such methods would reflect the implicit uncertainty WILDEBEAST faces when choosing a run to report from, rather than just relying on estimates from one. These methods could easily be incorporated into WILDEBEAST due to the modularity of the system, but would require careful evaluation due to the stochastic and unpredictable nature of MCMC samples, especially in distributions that exhibit bi-modality.

Discussion

Overall, these results show that WILDEBEAST functions correctly, and can give meaningful parameter estimates even when runs have not yet converged. Table 8.1 shows the percentage of timesteps where reported and true estimates given through subsampling gave estimates that were closer to the gold standard estimate than estimates given without subsampling. The true best estimates reflect that WILDEBEAST operating with subsampling gives better estimates up to 79.36% during the period of early characterisation of a pathogen, and this result stands in stark contrast with results obtained in Chapter 6 - ie when looking over a short time span, subsampling methods give better estimates faster than the full dataset due to the slower mixing time in chains runs on larger datasets. In an environment when data arrives this quickly, and continues to arrive after the 120th timestep, subsampling would prove crucial.

For brevity, graphs for TMRCA and growth rate are omitted

Parameter	Setting 2 true	Setting 2 reported
Evolutionary rate	68.25%	50.79%
Growth rate	79.36%	35.26%

Table 8.1: Percentage of timesteps where Setting 2 (subsampling) performed better than Setting 1 (no subsampling)

Of concern is the accuracy of the reporting function of WILDEBEAST, through which quality of reported estimates is diminished. The second column of Table 8.1 reflects the fact that subsampling begins to perform worse than using the full dataset when considering reported estimates, and this is due to the fact that the current method of using realised potential does not favour newly started runs which when using subsampling tend to give better estimates faster. Hence, a simple adjustment of the realised potential function could improve results.

Table 8.2 presents, for each setting, the percentage of estimates at each timestep that were within 50 percent of the true best estimate for that setting at the timestep. These results are generally poor and, as mentioned above, reflect the need to smooth reported estimates to reduce the erratic jumps in reported values seen above. Suggested techniques for doing this have been discussed above, but overall these percentages show that WILDEBEAST fails to fully consider estimates from all runs when reporting, and extending the reporting function to do so would require simple modifications to the framework.

Parameter	Setting 2 reported	Setting 1 reported
Evolutionary rate	39.92%	29.62%
Growth rate	31.82%	61.90%

Table 8.2: Percentage of timesteps where the reported estimate for each setting was within 50 percent of the true best estimate at that step

8.2.2 Experiment 2: Unfiltered dataset

The same experiment was carried out as above, except the unfiltered dataset was used to generate weekly cumulative knowledge, and $vectorDist(1, 1, 1)$ used for 50 percent subsampling in Setting 2. The unfiltered dataset is a significant computational challenge, as it represents the largest set of sequence data from a single epidemic that occurred over a short time span (relative to for example, HIV-1). For example, 87 sequences appeared in the first week with filtered data, compared to 474 for unfiltered (See Figure 5.1). Filtering has been applied in previous studies to make run times tractable [18].

Evolutionary rate estimates

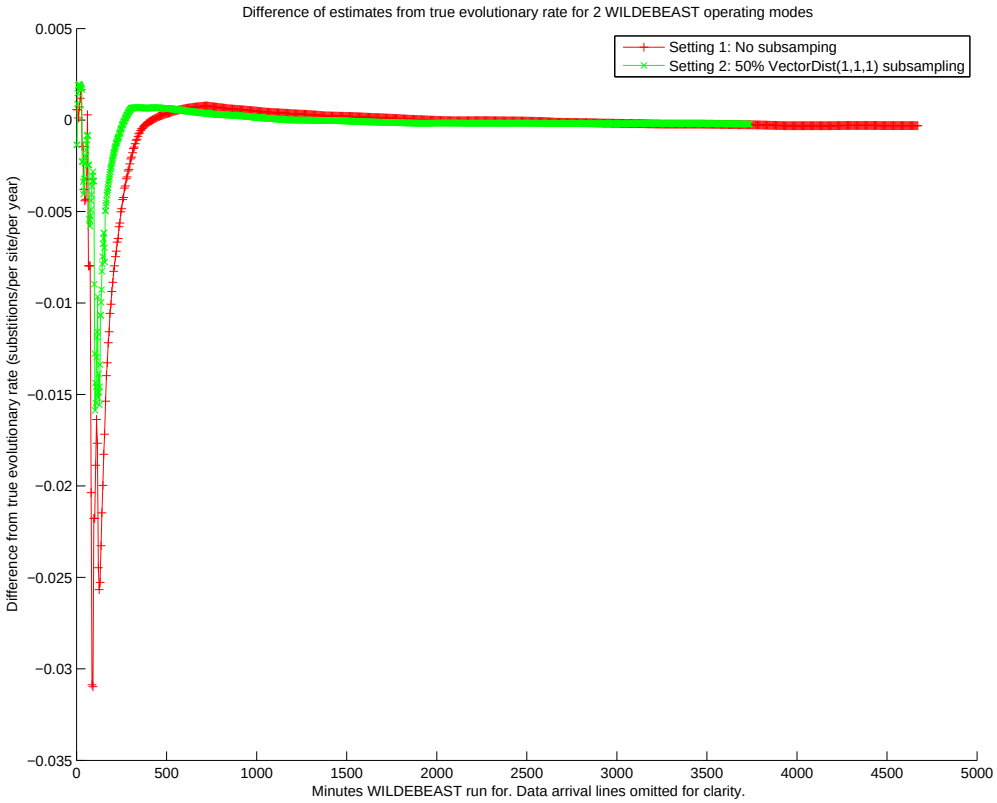
WILDEBEAST was left to run autonomously for approximately 70 hours with Setting 1, and 62 hours with Setting 2. Figure 8.3 a shows the $trueRate$ - $estimatedRate$ for each timestep, unlike before this is not an absolute distance. Figure 8.4 b shows the absolute distance in the reported estimates. Timesteps in both graphs are in real time minutes from the introduction of the first week of data. Data arrival lines are omitted for clarity.

Both plots show that WILDEBEAST was able to operate correctly over a long period on real world, unfiltered data, with no errors reported. Figure 8.3a shows that estimates given for the rate are very similar with both settings - even though Setting 2 only uses half the available data. This is evidence that the automated subset selection achieved through $vectorDist(1, 1, 1)$ is of great use in future data intensive epidemics.

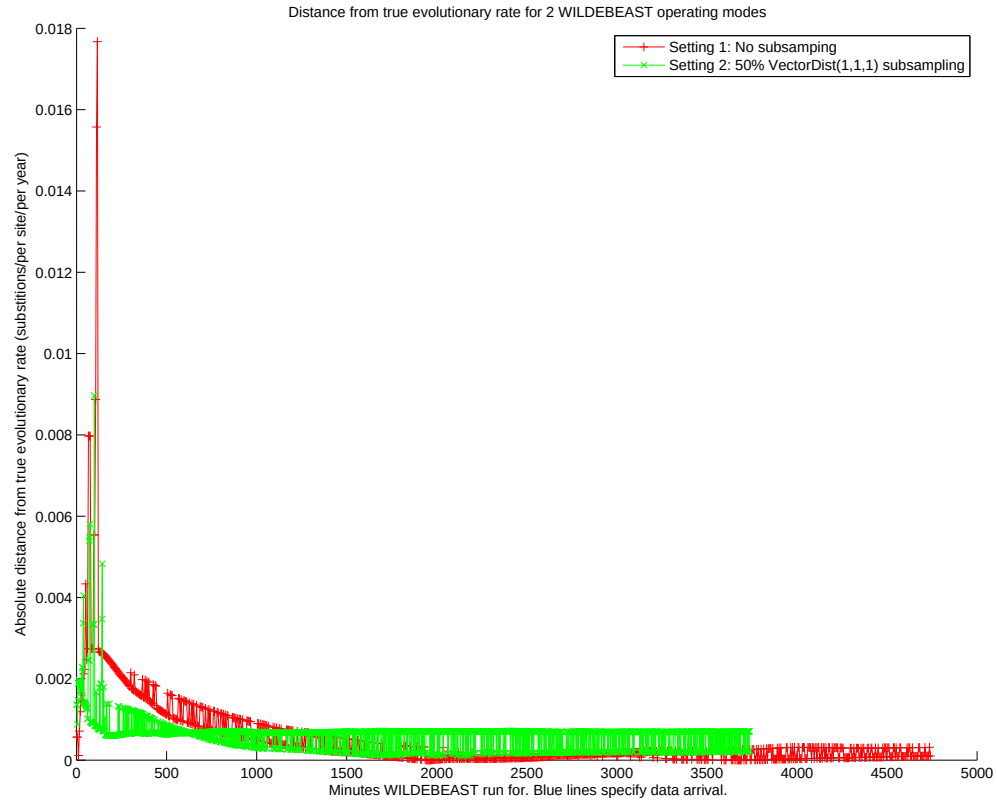
Figure 8.4 shows the same results, but plotted over 1000 minutes - the period in which early characterisation of H1N12009 would have happened. In addition, these plots show the arrival of new data, the values for the rate estimated by each setting, and the horizontal magenta line shows the gold standard rate.

Discussion

Figure 8.3b shows that in the long run, slightly better estimates are reported when using the full dataset about 1500 minutes into the experiment. Within 400 minutes of runtime, both settings are able to report results within 0.001 substitutions/site/year of the true result, and maintain these reports. The large deviations away from the true rate during the first 100 or so minutes are due to poor mixing in the chains - these deviations are more significant for Setting 1 as the unfiltered dataset runs take exceptionally long to accurate ESS - for example, the run started on the

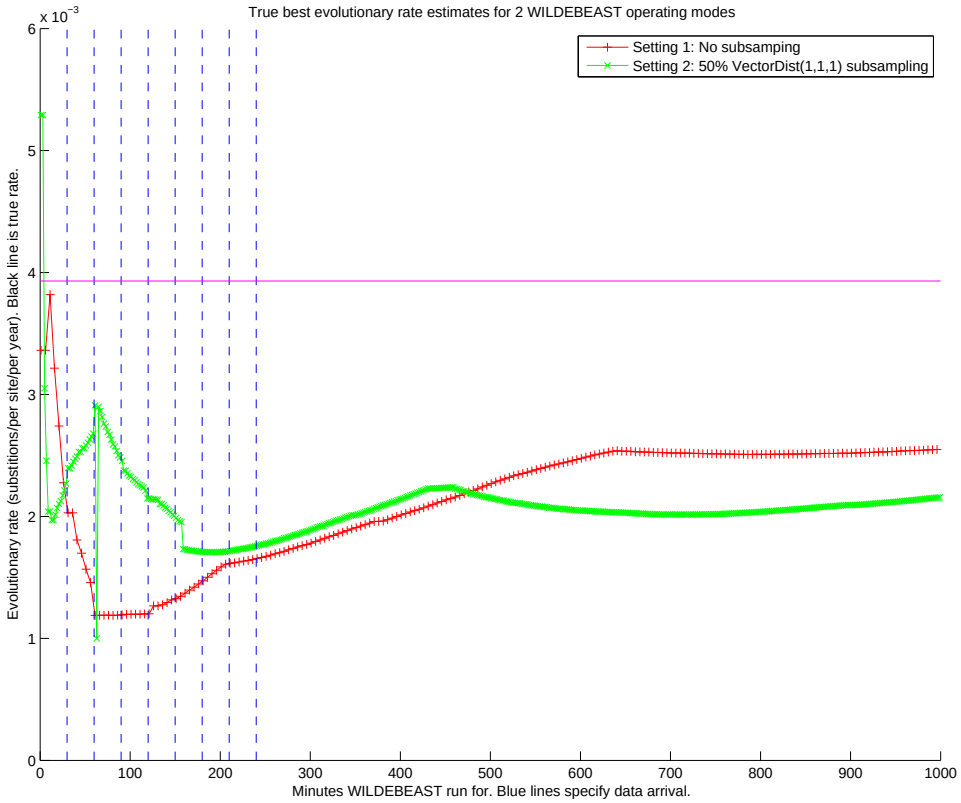


(a) Difference of true best estimate from gold standard evolutionary rate

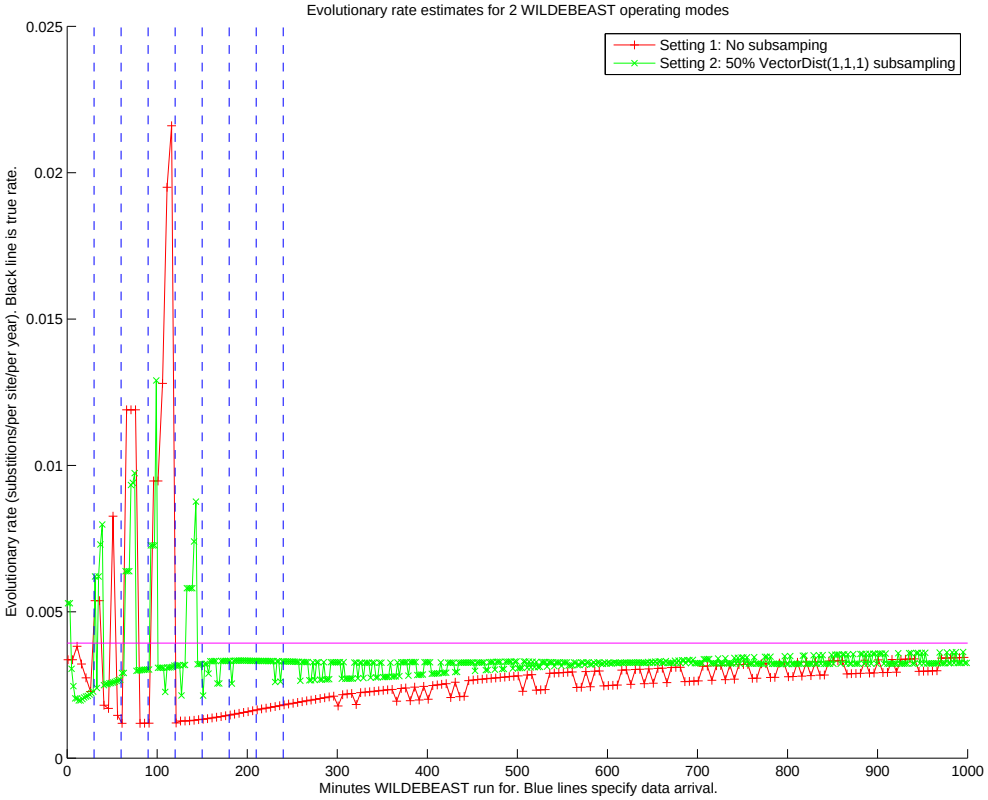


(b) Absolute distance of reported best estimate from gold standard evolutionary rate

Figure 8.3: Experiment two results over 5000 minutes



(a) True best estimates compared to gold standard evolutionary rate



(b) Reported best estimates compared to gold standard evolutionary rate

Figure 8.4: Experiment two results over 1000 minutes

first set of unfiltered data took over 3 days of runtime to accurate an ESS of 32 in the evolutionary rate, whereas even the largest filtered dataset accrued double this ESS in a shorter runtime. Table 8.5 compares the ESS of runs for each method. A second explanation for these large jumps may be that the 10% burn-in period that was defined for all runs (during which a run will not report estimates) was not large enough for these much larger datasets.

Figure 8.4a, which displays short-time behaviour of the system, shows that from the arrival of the second week of data until about the 450th minute, the best possible Setting 2 estimates are better than those given by Setting 1. However, Figure 8.4b shows that WILDEBEAST is able to report better estimates with subsampling from the introduction of the 2nd week of data to the end of the 1000 minute period. This happens because the filtered dataset runs accumulate ESS at a much faster rate due to a significantly lower dimension, and also mix faster, allowing WILDEBEAST to report better estimates with more confidence than with Setting 2. During the arrival of new data, estimates sometimes significantly overestimate the true rate, but a general trend that approaches the true value can still be seen, and smoothing methods discussed above could mitigate such jumps. Setting 2 deviates less than Setting 1 during this period, showing that filtering implemented by WILDEBEAST handles incoming data well. Percentage-based results for this period are discussed below.

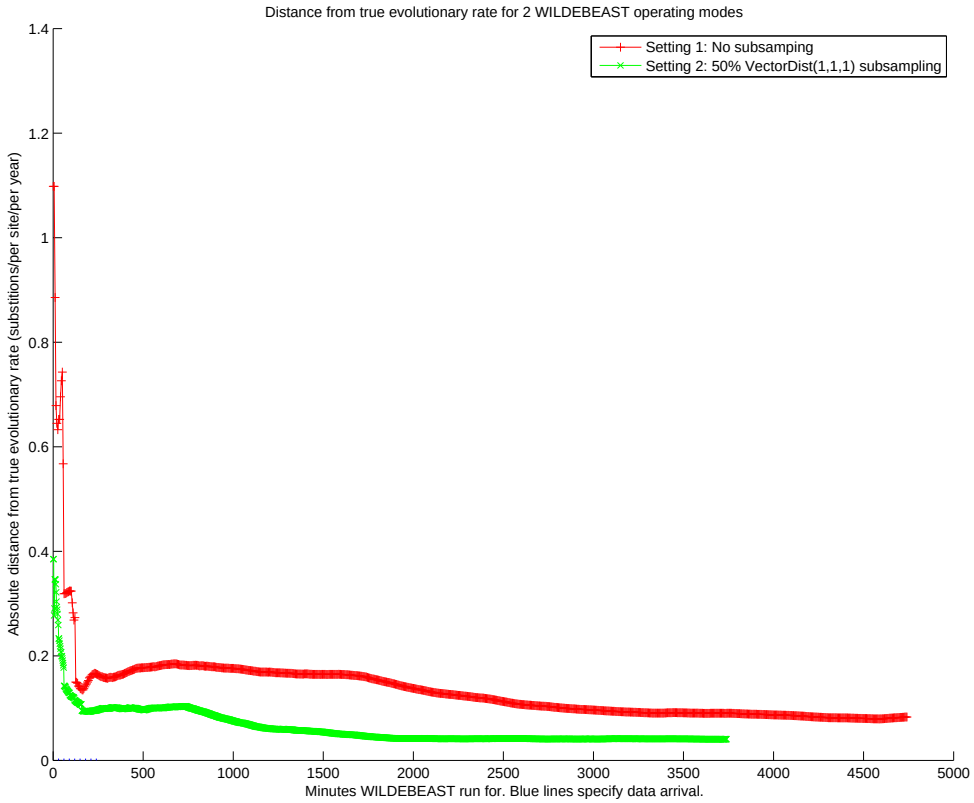
TMRCA estimates

The results for the TMRCA of the virus are plotted as absolute distance from the true TMRCA in Figure 8.5. Here, a much clearer split is seen between Setting 1 and 2 than with the evolutionary rate results, with Setting 2 always outperforming Setting 1, in both reported and true estimates. These results are also reflected in Table 8.4 below. This occurred because, as mentioned in Chapter 2, the hardest parameter to sample during a run is the phylogeny, and here TMRCA is represented by the height of the root node of such a sampled tree. Hence the unfiltered data simply takes much longer to give accurate estimates of TMRCA, given slow accumulation of ESS (see Table 8.5).

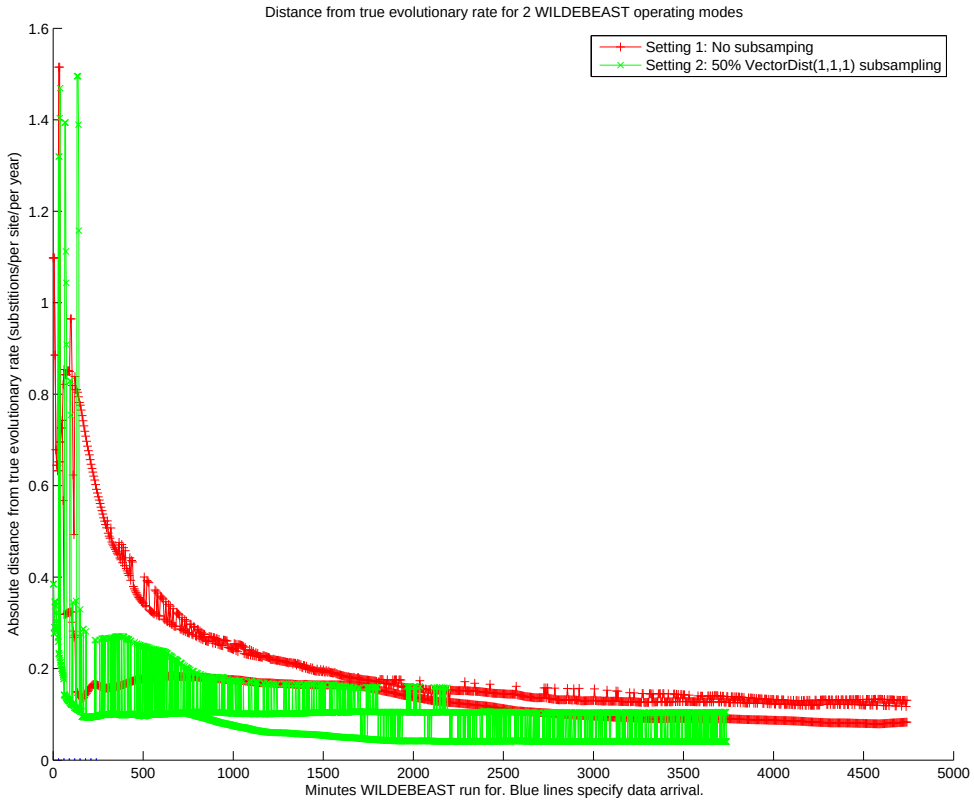
Discussion

Table 8.3 and 8.4 compare the performance of subsampling to non subsampling over both the first 500 and 1000 minutes. In both the first 500 and 100 minutes it is clear that subsampling is successful in giving faster TMRCA estimates. TMRCA estimates given with Setting 1 are up to half a year away from the true TMRCA during the first 500 minutes - or 30 weeks of a pandemic on our timescale - while Setting 2 reports estimates within 1 month of the true date during this phase. Getting accurate estimates of TMRCA earlier can be crucial in discovering the exact origins of the pathogen and containing further spread of a disease, which could prevent epidemics becoming pandemics and reduce fatalities on a global scale.

Both tables give better results for subsampling than those seen on the filtered dataset, which is promising as this dataset simulates a more realistic epidemic



(a) Distance of true best estimate from gold standard TMRCA



(b) Distance of reported best estimate from gold standard TMRCA

Figure 8.5: Experiment two results over 1000 minutes

Parameter	Setting 2 true	Setting 2 reported
Evolutionary rate	68.4%	84.4%
TMRCA	100%	96.4%
Growth rate	48.4%	93.2%

Table 8.3: Percentage of timesteps where Setting 2 (subsampling) performed better than Setting 1 (no subsampling) over the first 500 minutes

environment. Subsampling preforms better than the full dataset when estimating growth rate over 1000 minutes rather than 500, and worse for the evolutionary rate. On inspection of graphs (here omitted), growth rate estimates are very variable for both settings during the early stages of the epidemic.

Parameter	Setting 2 true	Setting 2 reported
Evolutionary rate	62.2%	45.4%
TMRCA	100%	93.8%
Growth rate	74.6%	95.8%

Table 8.4: Percentage of timesteps where Setting 2 (subsampling) performed better than Setting 1 (no subsampling) over the first 1000 minutes

The reported estimates with Setting 2 are higher than the filtered set due to the fact that ESS plays a greater role in distinguishing the quality of runs. Table 8.5 shows the ESS accumulated in the TMRCA, growth, and evolutionary rate for the first 6 runs under each. All of these results show that methods implemented in WILDEBEAST are useful features for early characterisation of a pathogen, and give competitive results when compared to using the full dataset.

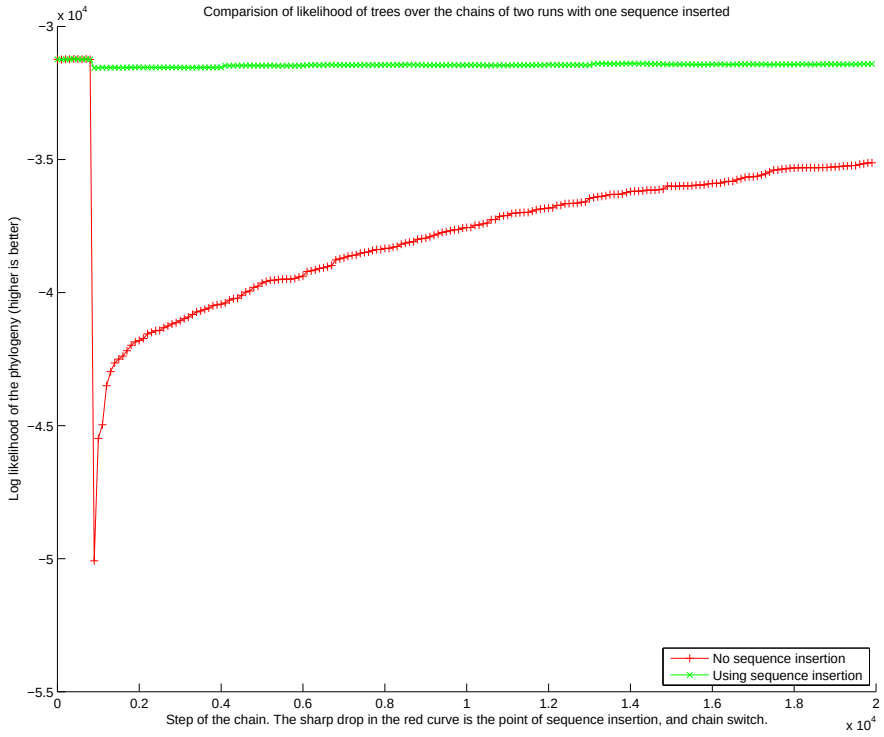
Run	Setting 1	Setting 2
Week 1	5215.0372	777.5391
Week 2	659.3538	182.4532
Week 3	388.3852	117.414
Week 4	179.1479	44.2847
Week 5	224.0035	35.1024
Week 6	33.9605	26.5408

Table 8.5: ESS in parameters of interest after 62 hours for Setting 2 and 70 hours for Setting 1

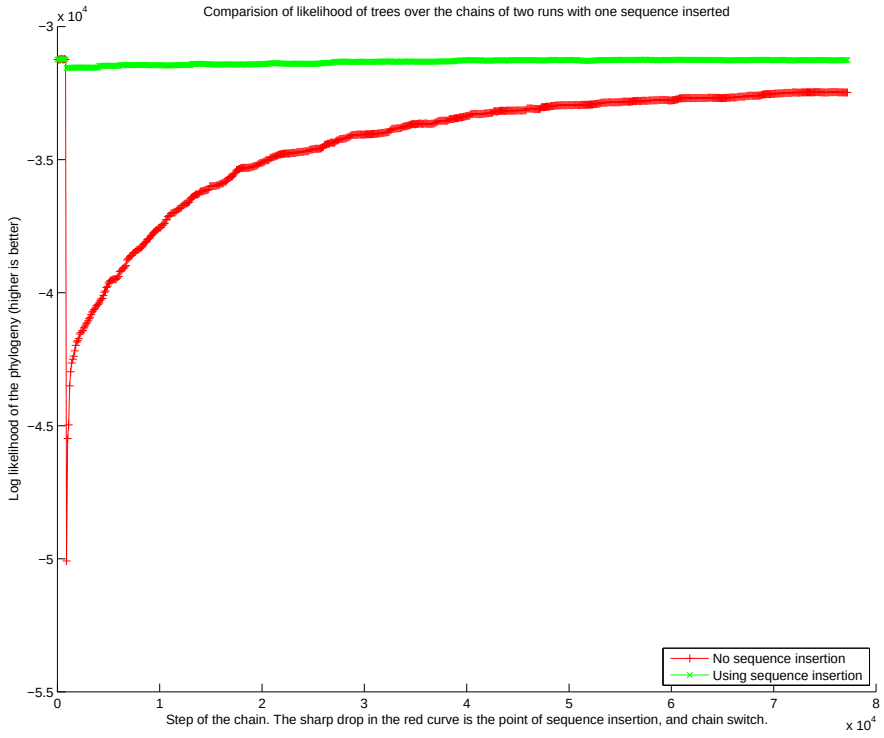
8.2.3 Sequence addition evaluations

Proof of concept

As an implementation of the sequence addition algorithm is non-trivial, a proof of concept was first carried out manually. An analysis with 207 sequences extracted from the first half of H1N12009 was run for 6257000 steps, and achieved over 500 ESS in each parameter of interest. A single new sequence was selected from the



(a) Likelihood of tree over 20000 chain steps



(b) Likelihood of tree over 80000 chain steps

Figure 8.6: Short and long term likelihood behaviour with and without sequence insertion

remaining H1N12009 data and the sequence insertion algorithm carried out to start a new run using the results of the previous run. Trees were inspected manually to ensure the sequence was inserted correctly. Figure 8.6 shows the likelihood over the phylogeny plotted for both the previous run and new run over the insertion of the sequence using either sequence insertion, or the standard BEAST method. Figure 8.6a shows the short term behaviour (20 thousand steps) and 8.6b long term (80 thousand steps). Sequence insertion performed significantly better when handling the addition of a new sequence, as the likelihood of the tree only drops a fraction compared to without sequence insertion. It should be noted that burn-in periods are not shown on these graphs yet even after 80000 steps the run without sequence insertion has yet to reach the likelihood that the sequence insertion method has been sampling at for the entire period - allowing it to accrue higher ESS and report more reliable parameter estimates faster.

Sequence size	With insertion	Without insertion
208	96	79
209	94	69
210	125	98
211	100	44
212	160	65
213	145	130

Table 8.6: Rounded ESS in parameters of interest over full 1000000 chain length for sequence insertion versus control

Autonomous addition

Sequence insertion was implemented into the automated process of WILDEBEAST. 6 sequences were randomly selected from H1N12009 as the new sequence set. The results of the previously described 207 sequence run were added to WILDEBEAST, and it was left to run autonomously while each of the 6 sequences were introduced individually over time. The results for the first 4 sequences are plotted below in Figure 8.7. It is clear that sequence insertion is a powerful algorithm - indeed if the arrival of sequences is rapid, such as the gap between the insertion of sequence 1 and 2 in this insertion, not using sequence insertion and only maintaining one run will result in chains never being able to mix properly. The stable likelihood which the sequence inserted runs maintained throughout the period is promising, and Table 8.6 confirms that chains started with sequence insertion were consistently able to mix more than those without out a fixed chain length, resulting in better approximations of evolutionary parameters.

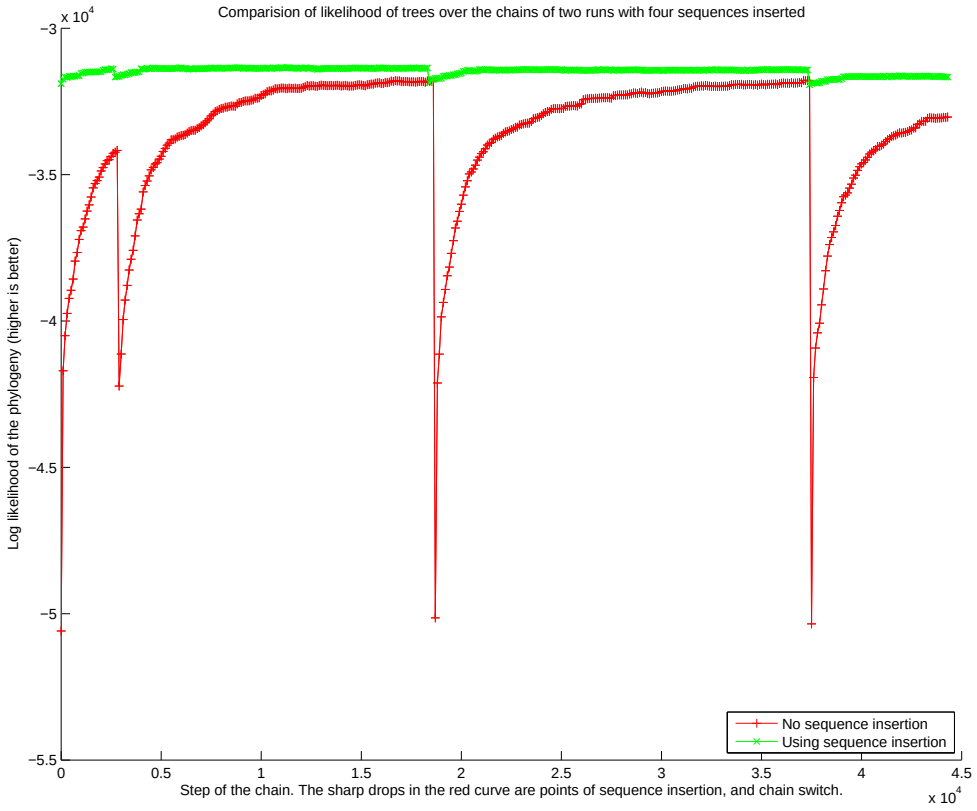


Figure 8.7: Likelihood of tree - sequence addition vs no sequence addition, both over four runs/insertions

8.3 Other epidemic evaluations

A key feature of WILDEBEAST is its ability to be deployed for use in future epidemics. To evaluate if the system was able to generalise, a number of experiments were carried out on sequence data from other epidemics introduced in Chapter 5. The gold standard estimates for evolutionary parameters were retrieved from a number of sources, and are summarised in table 8.7

Epidemic	TMRCA	Evolutionary Rate	Growth Rate	Source
H3N2	2002.5	0.0572	N/A	[35]
DENV-1	N/A	0.009688	0.1729	Run on full data
SARS	0.001456	N/A	0.6465	Run on full data

Table 8.7: Gold standard evolutionary parameter estimates per pathogen and their sources

8.3.1 H3N2

Sequences in the H3N2 epidemic span from 2003.019 to 2005.979. On 2003.44, 19 sequences existed, and this was chosen at the starting point for analysis with cumulative datasets generated every month for 11 months - up to 2004.396, by which 239 sequences were available. Each monthly dataset was uploaded every 30 minutes.

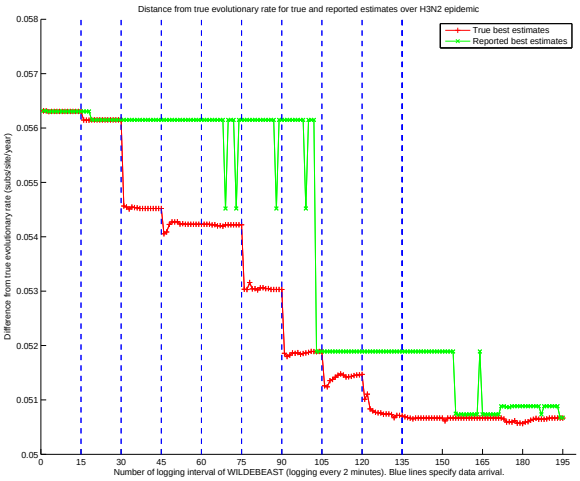
Figure 8.6a shows the reported vs actual best estimates at each timestep for the evolutionary rate, and 8.6b for the TMRCA. Subsampling was not applied as the datasets are small, and all runs achieved over 300 ESS in the relevant parameters. These results show that the reported best estimates between timestep 30 and 105 performed poorly and failed to select runs with better estimates to report from. This is due to the fact since datasets were relatively small, new runs gave better estimates quicker, but the $spread \times ESS$ rule preferred older runs as they had higher ESS. This implies that this rule does not work well in all cases and additional reporting rules need to be considered. Fom the 150th timestep onwards reported TMRCA values were very close to the true TMRCA, and this is a good result.

8.3.2 DENV-1

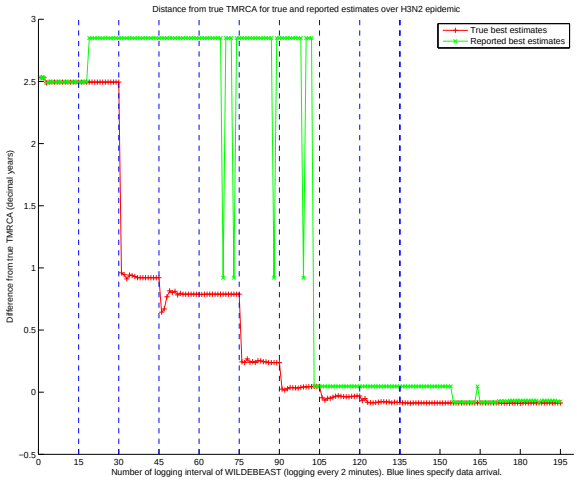
DENV-1 was broken up into 24 monthly datasets , the first spanning from 2003.691 to 2006.187 (38 sequences), and final 2003.691 to 2008.13 (560 sequences), uploaded every 30 minutes. This test was carried out to evaluate the phasing system and performance with strict limitations - $concurrent_{limit}$ was set to two runs at a time, and $runtime_{limit}$ to 4 hours. Figure 8.8c and 8.8d display the distance of estimates from gold standard values, for the last 16 data arrivals. The system correctly stopped runs with lowest realised potential to spawn new ones, but the rapid rate at which data arrived combined with run halting means that evolutionary rate estimates did not consistently improve with the introduction of more data. That happens for two reasons, firstly, runs were stopped too quickly to allow them to mix (all runs were stopped before they met the minimum ESS of 300), and secondly, the system began to downsample every cumulative dataset to 165 sequences, as warned earlier, meaning the difference in the datasets of latter runs constitutes only a few sequences, analysis of which fail to give better estimates in short timespans. WILDEBEAST correctly flagged the epidemic as being in phase three when this began to happen, as human intervention was needed since WILDEBEAST, like BEAST, can fail if operational parameters are not set carefully. Either concurrent run or run time restrictions need to be relaxed by as sequence data increases, and the phasing system can help humans do this.

8.3.3 SARS

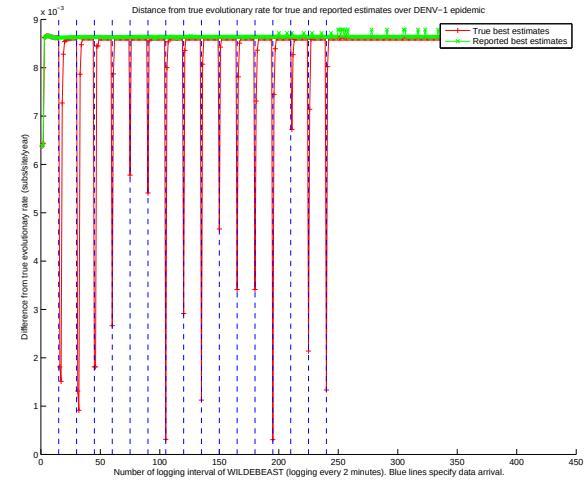
The total SARS dataset contains only 73 sequences . Looking at monthly intervals starting on 2003.13, 35 sequences are available with the first month, 13 more in the next month, and then an absence of new sequences for 7 months. 13 monthly sequence sets were uploaded every 30 minutes, including sets that have no change



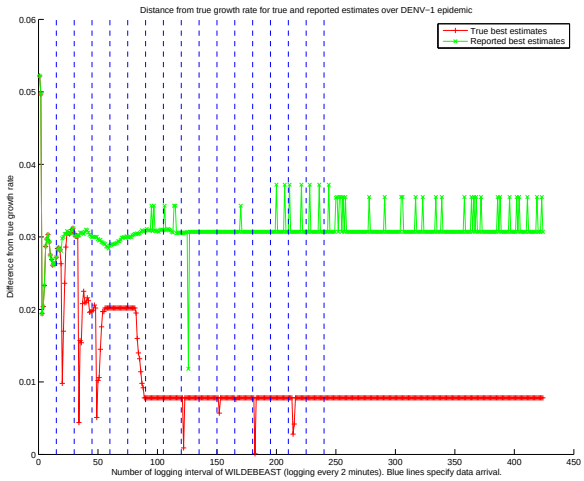
(a) H3N3: Evolutionary rate estimates



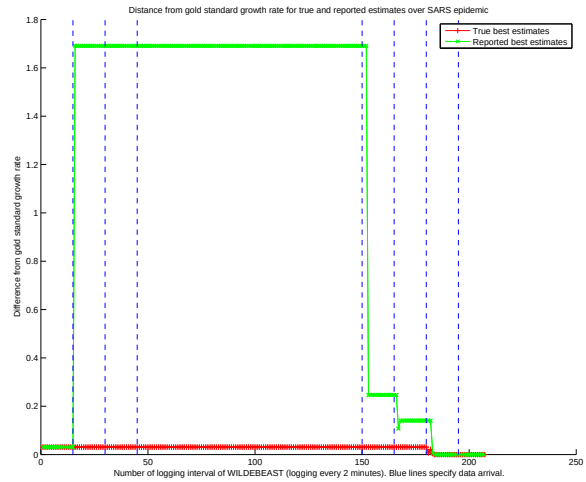
(b) H3N2: TMRCA estimates



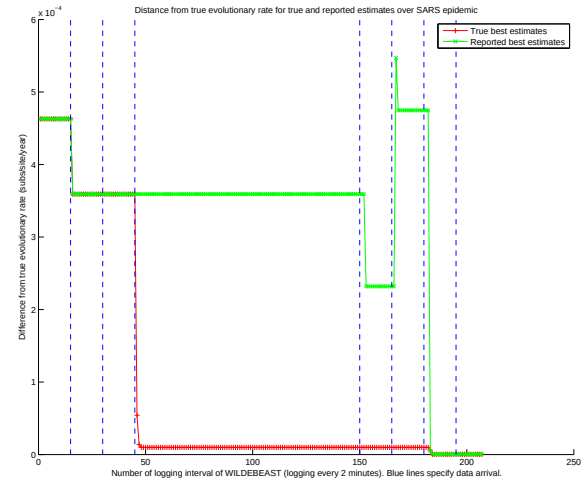
(c) DENV-1: Evolutionary rate estimates



(d) DENV-1: Growth rate estimates



(e) SARS: Growth rate estimates



(f) SARS: Evolutionary rate estimates

Figure 8.8: WILDEBEAST evaluations on DENV-1, SARS, and H3N2

from the previous month. As data is sparse in this epidemic, run threshold parameters were relaxed. WILDEBEAST correctly classified this epidemic as being in phase 1 .

Figure 8.8e and f shows that during the period of no new sequences, WILDEBEAST failed to update reported estimates when a better result was available. This is because chains were run for a fix length - if they have been run for very long periods, eventually the most recent dataset would have achieved higher ESS than previous sets, and since they have a higher spread, would have had the highest realised potential. This infinite chain length system was not implemented for practical purposes, but in data-constrained settings such as this, the system needs to be adjusted to make best use of the data that has been seen if sequences aren't arriving at a rapid rate. This could possibly be implemented as an extension to the phasing system, for Phase 1 epidemics.

Chapter 9

Conclusions

The principal goal of this project was to build a system for real time genetic analysis of infectious disease epidemics. This was achieved through the implementation of the WILDEBEAST web service, which integrated a number of novel methods that have been shown to work cohesively to overcome challenges in the process of real time characterisation of viral pathogens.

9.1 Future directions

The retrieval of sequences from public repositories would be a useful addition to WILDEBEAST, and would be simple to integrate into the provided framework. A module independent from the WILDEBEAST controller could run on a server to query a set of sequence repositories, parse data formats appropriately, and update the $newSeqs.fasta$ file for each epidemic.

A number of minor features were not implemented due to time limitations. This includes functionality to automatically generate graphs of evolutionary estimates over time, an interface for users to upload evolutionary model templates for particular pathogens, a login system restricting navigation beyond the summary page to authenticated researchers, and automatic feedback of the learning component into future predictions.

The framework allows extension in a number of ways, such as the addition of more sequence selection algorithms, more robust computations or algorithms for sequence insertion, and the introduction of new reporting mechanisms, for example, measuring potential and realised potential by a more complex weighting of run features, and smoothing parameter reports by looking at more than just one run. Future work cloud also focus on improving the ESS per step of BEAST by adjusting proposal distributions on the fly for efficiency.

9.1.1 Originality in this project

WILDEBEAST is the first system to carry out real time analysis of sequence data in order to infer evolutionary estimates of epidemics. Other real time epidemic

monitoring systems do not make use of molecular sequence data, and are not able to give estimates of evolutionary parameters through cutting-edge Bayesian phylogenetic techniques.

Chapter 3 and 4 show that the system is immediately available for use through deployment on a server, and has been developed to a standard that is more than just a proof of concept. The system caters both for the general public, experienced researchers, and operational parameters allow significant flexibility in its operation. Code for the project has been released into the open source community along with a wiki, and it is likely that there will be further development of the system, given the modular nature of the system.

The work presented in Chapter 5 constitutes the first attempts at predicting the performance of BEAST analysis from features of viral sequences. The data dimension quantity was discovered to be highly predictive of ESS per hour and other features of a run, even holding across different types of sequence data. These findings allow users and systems to make informed decisions about starting BEAST runs, an effectively allow evolutionary epidemiologistcs to carry out timely reporting of parameter estimates to policy makers.

Chapter 6 presents the first formal study of filtering techniques in the context of epidemics - an important problem given ever amassing quantities of sequence data, and the fact that BEAST analysis are exponential in sequence size. The $vectorDist$ algorithm is the first flexible solution to this problem, and an improvement of ad hoc methods applied for filtering in previous literature.

The sequence insertion algorithm discussed in Chapter 7 constitutes a powerful solution to the conceptual problem of continuously arriving sequence data, and gives a significant improvement over the current method for inserting sequences. It is possible that this method could be integrated into the BEAST software package, along with additional enhancements.

The concepts of potential and realised potential are a step in the right direction for comparing running BEAST analysis, and have allowed creation of a unique decision process that is able to reason about MCMC, and ultimately allows BEAST to function in a real time setting.

The system has been evaluated on real data from a diverse set of past epidemics. WILDEBEAST was able to carry out real time characterisation of H1N12009, using the largest collection of sequence data from a single epidemic over a short span, and such performance proves the usefulness of the system.

Overall, the author believes this project has advanced the field of epidemic monitoring and characterisation, and produced a system that extends BEAST in a way that now makes Bayesian phylogenetic techniques feasible for real time study of pathogens. The author intends to submit an applications note for peer review, as a means of alerting the research community about the system, and believes WILDEBEAST will be deployed in the near future for real world use.

9.2 Conclusion

In the 1954 fiction novel ‘I Am Legend’, protagonist Robert Neville is the only human left on Earth after a viral pathogen induces vampirism in the remaining population. Captured and held for execution by these new sentient beings, he reflects on the fact that the terror he once held for the pathogen is insignificant when compared to the terror that the new populace of the Earth holds for him, given that he has hunted their brothers and sisters. Robert finds solace in this, as he believes that even in death, he will be immortalised as “a new terror born in death, a new superstition entering the unassailable fortress of forever." [33]

On our Earth, it is viral pathogens that stoically persist in the unassailable fortress of forever, as neither living nor dead. Our ancestors regarded these maladies as the work of some manifestation of legend - an unseen superstition which left in its wake only death and terror. Today, this terror is diminished only through understanding; understanding that is championed by tools such as WILDEBEAST and BEAST .

Bibliography

[1] Boccaccio, Giovanni. 1930, *The Decameron Vol. I (translated by Richard Aldington, illustrated by Jean de Bosschere)*, Published by Filippo and Bernardo Giunti.

[2] Bos et al. 2001, *A draft genome of Yersinia pestis from victims of the Black Death*, Nature 478, 506-509 (doi:10.1038/nature10549)

[3] Wallis, E. A. 1904, *Gods of the Egyptians*, Gilbert and Rivington.

[4] Darwin C. 1859, *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*, Nature 5, 318-319 (doi:10.1038/005318a0).

[5] Yang Z et al, 2012, *Molecular phylogenetics: principles and practice*, Nature Reviews Genetics 13, 303-314 (doi:10.1038/nrg3186).

[6] Lemey P, Salemi M, Vadamme E, 2009, *The Phylogentic Handbook: a Practical Approach to Phylogentic Analysis and Hypothesis Testing*, Cambridge University Press.

[7] Drummond A, Rambaut A. 2007, *BEAST: Bayesian evolutionary analysis by sampling trees*, BMC Evolutionary Biology 2007, 7:214 (doi:10.1186/1471-2148-7-214).

[8] Grenfell B et al. 2004, *Unifying the Epidemiological and Evolutionary Dynamics of Pathogen*, Science 303, 327. (doi: 10.1126/science.1090727)

[9] Pybus OG, Fraser C, Rambaut A. 2013, *Evolutionary epidemiology: preparing for an age of genomic plenty*, Phil Trans R Soc B 368: 20120193. (doi:10.1098/rstb.2012.0193)

[10] Colizz V, Oliveira T, Roberts R. 2007, *Libya should stop denying scientific evidence on HIV* Nature 448, 992. (doi:10.1038/448992)

[11] Oliveira et al. 2006, *Molecular Epidemiology: HIV-1 and HCV sequences from Libyan outbreak* Nature 444, 836-837. (doi:10.1038/444836a)

[12] Hughes GJ, Fearnhill E, Dunn D, Lycett SJ, Rambaut A, et al. 2009, *Molecular Phylodynamics of the Heterosexual HIV Epidemic in the United Kingdom*, PLoS Pathog 5(9): e1000590. (doi:10.1371/journal.ppat.1000590)

[13] Worobey et al. 2008, *Direct evidence of extensive diversity of HIV-1 in Kinshasa by 1960*, Nature 455. (doi:10.1038/nature07390) ï£ij

[14] Pybus O, Rambaut A. 2009, *Evolutionary analysis of the dynamics of viral infectious disease*, Nature Reviews Genetics 10, 540-550. (doi:10.1038/nrg2583)

[15] Cotten et al. 2013, *Transmission and evolution of the Middle East respiratory syndrome coronavirus in Saudi Arabia: a descriptive genomic study*, The Lancet, Volume 382, Issue 9909, 1993 - 2002. (doi:10.1016/S0140-6736(13)61887-5)

[16] Smith G et al. 2009, *Origins and evolutionary genomics of the 2009 swine-origin H1N1 influenza A epidemic*, Nature 459, 1122-1125. (doi:10.1038/nature08182)

[17] Fraser et al. 2009, *Pandemic Potential of a Strain of Influenza A (H1N1): Early Findings*, Science 324, 1557-1561. (doi:10.1126/science.1176062)

[18] Hedge J, Lycett SJ, Rambaut A. 2013, *Real-time characterization of the molecular epidemiology of an influenza pandemic*, Biol Lett 9: 20130331. (doi:10.1098/rsbl.2013.0331)

[19] Dawood F et al. 2012, *Estimated global mortality associated with the first 12 months of 2009 pandemic influenza A H1N1 virus circulation: a modelling study*, The Lancet Infectious Diseases Vol. 12, Issue 9, 687-695. (doi:10.1016/S1473-3099(12)70121-4)

[20] Rambaut A. 2009, *Human/Swine A/H1N1 Influenza Origins and Evolution*, Accessible at http://tree.bio.ed.ac.uk/groups/influenza/

[21] World Health Organization. 2009, *Global surveillance during an Influenza pandemic*, Accessible at `http://www.who.int/csr/disease/swineflu/global_pandemic_influenza_surveilance_apr09.pdf`

[22] World Health Organization, 2009, *Pandemic Influenza Preparedness and Response: A WHO Guidance Document. Geneva: ; 4, THE WHO PANDEMIC PHASES.* Accessible at `http://www.ncbi.nlm.nih.gov/books/NBK143061/`

[23] Butte A. 2008, *Translational Bioinformatics: Coming of Age*, J Am Med. Inform. Association 2008 15:709-714. (doi: 10.1197/jamia.M2824)

[24] Rambaut A et al. 2004, *The causes and consequences of HIV evolution*, Nat. Rev. Genet., 5 (2004), pp. 52âĂŞ61. (doi: 10.1038/nrg1246)

[25] Drummond et al. *Inference of Viral Evolutionary Rates from Molecular Sequences*

[26] Jukes TH, Cantor CR. 1969, *Evolution of Protein Molecules*, New York: Academic Press. pp. 21-132.

[27] Kimura M. 1980, *A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences*, Journal of Molecular Evolution 16 (2): 111âĂŞ120. (doi:10.1007/BF01731581)

[28] Drummond A, Ho S, Phillips M, Rambaut A. 2006, *Relaxed Phylogenetics and Dating with Confidence*, PLoS Biol. May 2006; 4(5): e88. (doi: 10.1371/journal.pbio.0040088).

[29] Hrbek T, da Silva VMF, Dutra N, Gravena W, Martin AR, et al. 2014, *A New Species of River Dolphin from Brazil or: How Little Do We Know Our Biodiversity.* LoS ONE 9(1): e83623. (doi:10.1371/journal.pone.0083623)

[30] Orlando et al. 2013, *Recalibrating Equus evolution using the genome sequence of an early Middle Pleistocene horse*, Nature 499, 74-78 (doi:10.1038/nature12323).

[31] Alex Bielovich, 2014, *WILDEBEAST logo* `http://alexanderblv.co.za/`

[32] Hohna S, Drummond A. 2010, *Guided tree topology proposals for Bayesian Phylogenetic Inference*, Syst Biol (2012) 61 (1): 1-11 (doi: 10.1093/sysbio/syr074).

[33] Matheson, Richard. 1926, *I Am Legend* New York, Fawcett Publications [1954] (OCoLC)654725572, ISBN: 9782207300107

[34] Raghwani J, Rambaut A, Holmes EC, Hang VT, Hien TT, et al. 2001, *Endemic Dengue Associated with the Co-Circulation of Multiple Viral Lineages and Localized Density-Dependent Transmission*, PLoS Pathog 7(6): e1002064. (doi: 10.1371/journal.ppat.1002064)

[35] Salemi et al. 2004, *Severe Acute Respiratory Syndrome Coronavirus Sequence Characteristics and Evolutionary Rate Estimate from Maximum Likelihood Analysis* J. Virol. February 2004 vol. 78 no. 3 1602-1603 (doi: 10.1128/JVI.78.3.1602-1603.2004)

[36] Rambaut et al. 2008, *The genomic and epidemiological dynamics of human influenza A virus.*, Nature, 453(7195):615-9. (doi: 10.1038/nature06945)

[37] Carneiro et al, 2009. *Google Trends: A Web-Based Tool for Real-Time Surveillance of Disease Outbreaks* Clin Infect Dis. (2009) 49 (10): 1557-1564. (doi: 10.1086/630200)

[38] Cotten et al. 2013, *Transmission and evolution of the Middle East respiratory syndrome coronavirus in Saudi Arabia: a descriptive genomic study*, The Lancet. (http://dx.doi.org/10.1016/ S0140-6736(13)61887-5)

[39] David Lazer, Ryan Kennedy, Gary King, and Alessandro Vespignani. 2014, *The Parable of Google Flu: Traps in Big Data Analysis* Science: 343 (6176), 1203-1205. (doi:10.1126/science.1248506)

[40] Jenkins Gm, Rambaut A, Pybus OG, Holmes EC, *Rates of molecular evolution in RNA viruses: a quantitative phylogenetic analysis.* J Mol Evol. 2002 Feb;54(2):156-65.

[41] Hulsenbeck J, Ronquist F, 2001. *MRBAYES: Bayesian inference of phylogenetic trees*, Bioinformatics (2001) 17 (8): 754-755. (doi: 10.1093/bioinformatics/17.8.754)

[42] Hasegawa M, Kishino H, Yano T. 1985, *Dating of the human-ape splitting by a molecular clock of mitochondrial DNA*. J. Mol. Evol. 22 (2): 160âĂŞ74. doi:10.1007/BF02101694. PMID 3934395.