HOKKAIDO UNIVERSITY

Title	The Analysis of Infectious Diseases via Machine Learning
Author(s)	TESSMER, HEIDI LYNN
Degree Grantor	北海道大学
Degree Name	博士(獣医学)
Dissertation Number	甲第13258号
Issue Date	2018-06-29
DOI	https://doi.org/10.14943/doctoral.k13258
Doc URL	https://hdl.handle.net/2115/71247
Type	doctoral thesis
File Information	Heidi_Lynn_TESSMER.pdf

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

The Analysis of Infectious Diseases via Machine Learning

(機械学習による感染症の解析)

Heidi Lynn Tessmer

Abstract

This thesis introduces two projects applying machine learning methods to the realm of bioinformatics. In Chapter 1, we look at a regression problem involving the parameter values associated with the SEIR epidemiological model while in Chapter 2 we explore viral host classification.

Chapter 1 - To estimate and predict the transmission dynamics of respiratory viruses, the estimation of the basic reproduction number, R_0, is essential. Recently, approximate Bayesian computation methods have been used as likelihood free methods to estimate epidemiological model parameters, particularly R_0. In this paper, we explore various machine learning approaches, the multi-layer perceptron, convolutional neural network, and long-short term memory, to learn and estimate the parameters. Further, we compare the accuracy of the estimates and time requirements for machine learning and the approximate Bayesian computation methods on both simulated and real-world epidemiological data from outbreaks of influenza A(H1N1)pdm09, mumps, and measles. We find that the machine learning approaches can be verified and tested faster than the approximate Bayesian computation method, but that the approximate Bayesian computation method is more robust across different datasets.

Abstract

Chapter 2 - Infectious diseases which transfer between species are particularly difficult to manage. Knowing the natural host for an infectious agent makes it easier to prevent interspecies transmissions. However, with new and re-emerging disease, it can be difficult to know what the reservoir host is. In the second half of this thesis, we conducted a principal component analysis using data from the fruit bat and wild duck, along with a selection of single-stranded RNA viruses found in each animal. Historically, the virus-host relationship has often been examined using two components, that is, the G+C content of the genomes and the rate ratio of CpG in the genome. However, numerous data discrepancies exist which cannot be explained with mathematical models built from this technique. In this study, we found several alternative components that could be used to infer the host animal species of RNA viruses. Using these alternative components, we may be able to build a mathematical model that more closely simulates the virus-host genetic relationship. With this information, we may be able to identify genetic signatures in viruses which can uniquely identify the natural host species. In future, this information could help identify the animal source of a new outbreak.

Acknowledgements

I would like to acknowledge the many people who have supported me over the years and encouraged me to keep moving forward.

First, I would like to thank Professor Ryosuke Omori for the many hours of discussion in person, over email, and via Skype. Your patience, support, and high standards ensured the success of our many projects and my eventual graduation.

Next, I would like to appreciate Professor Kimihito Ito for encouraging me to find a topic I was sincerely interested in.

Thank you to Elizabeth Tasker, Bongkot Soonthornsata, Wallaya Phongphaew, Wessam Mohamed, and Gabriel Gonzalez for your friendship and advice. I enjoyed the many hours we spent in discussion and the encouragement I received from all of you.

I am grateful to my wonderful friends in Korea: Serim Jang, Lenin Gurung, and Ceyda Cinarel. Thank you for our many adventures and all the wonderful times we had.

My thanks go to the members of the Division of Bioinformatics at Hokkaido University and the members of the Biointelligence Laboratory at Seoul National University. It has been a pleasure working with all of you.

Acknowledgements

I would like to thank Greg Kipe for encouraging me to 'go for it' all those years ago. I am also grateful to Professor Michael Wick for his inspired lectures and enthusiasm for the field of computer science which has maintained my excitement in the field these past 18 years. To my teachers Randal Meinen and Kay Ziegahn who inspired me to work hard and challenge myself in any pursuit. I would also like to acknowledge Tracy Irving for her years of friendship and support.

Finally, I would like to thank my family for their love and support over the years.

Abbreviations

ABC: approximate Bayesian computation

A+U: the sum of A and U nucleotides

CNN: convolutional neural network

G+C: the sum of G and C nucleotides

LSTM: long short term memory

MERS: Middle East Respiratory Syndrome

ML: machine learning

MLP: multilayer perceptron

NCBI: National Center for Biotechnology Information

PCA: principal component analysis

RNN: recurrent neural network

SARS: Severe Acute Respiratory Syndrome

SEIR: Susceptible - Exposed - Infectious - Removed

SIR: Susceptible - Infectious - Removed

XpY: for any dinucleotide, this means the nucleotide base X preceding the nucleotide base Y

Table of contents

Abstract **iii**

Acknowledgements **v**

Abbreviations **vii**

List of figures **xiii**

List of tables **xv**

Preface **xvii**

1 Chapter 1 **1**

1.1 Introduction 1

1.2 Materials and Methods 5

1.2.1 Terminology 6

1.2.2 SEIR Epidemiological Model 7

1.2.3 Approximate Bayesian Computation 9

1.2.4 Machine Learning 10

1.2.5 Datasets 13
1.2.6 Time Calculations 14
1.2.7 Experiments on Real-World Datasets 15
1.3 Results 16
1.3.1 Comparisons of Average Errors 16
1.3.2 Comparisons of Credible / Confidence Intervals 18
1.3.3 Comparisons of Estimated and Actual Values 19
1.3.4 Run Times 21
1.3.5 Application to Real-World Epidemiological Data 22
1.4 Discussion 24

2 Chapter 2 **29**

2.1 Introduction 29
2.2 Materials and Methods 32
2.3 Results 34
2.4 Discussion 34
2.5 Conclusion 38
2.6 Accession Numbers 39

3 Conclusion **49**

Publications **51**

Presentations **53**

References **55**

List of figures

1.1 Average error for ABC (solid line), CNN (long dashed line), MLP (dotted line), MLP with time (dashed line), and LSTM (dashed and dotted) estimates against actual parameter values for an SEIR model with three parameters, on 100,000 training datasets. 17

1.2 Average error for ABC (solid line), CNN (long dashed line), MLP (dotted line), MLP with time (dashed line), and LSTM (dashed and dotted) estimates against actual parameter values for an SEIR model with three parameters, for one million training datasets. 18

1.3 Average credible/confidence interval width for ABC (solid line), CNN (long dashed line), MLP (dotted line), MLP with time (dashed line), and LSTM (dashed and dotted) estimates against actual parameter values for an SEIR model with three parameters, on 100,000 training datasets. 19

1.4 The estimated and actual parameter values from ABC, MLP, CNN, MLP with time, and LSTM. 20

List of figures

2.1 The principal component analysis has been conducted on all of the bat and duck genes and viruses, and the results displayed separated into the ‘Duck’ group (dark blue) and the ‘Bat’ group (light blue), with ellipses around the corresponding groups. 36

List of tables

1.1 The time required to train 100,000 samples and test on a single sample for the ABC, MLP, CNN, and LSTM methods. 21

1.2 The time required to train one million samples and test on a single sample for the ABC, MLP, CNN, and LSTM methods. 22

1.3 The estimation results for R_0 from previous published research on mumps and ABC, two MLP, CNN, and LSTM models. Estimates shown include the 95% credible/confidence intervals. 22

1.4 The estimation results for effective reproduction number from previous published research on measles and ABC, two MLP, CNN, and LSTM models. Estimates shown include the 95% credible/confidence intervals. 23

1.5 The estimation results for R_0 from previous published research on influenza and ABC, two MLP, CNN, and LSTM models. Estimates shown include the 95% credible/confidence intervals. 23

2.1 The nucleotide sequences of bat, bat virus, duck and duck viruses used in this study. 33

List of tables

2.2 Frequency of dinucleotides in bat, bat viruses, duck, and duck viruses. . . . 35

2.3 Percentages of variance explained by the principal components over all reads (bat, duck, and all bat and duck viruses). 37

Preface

Machine learning has emerged as an effective tool for the analysis of complex, non-linear data. In this thesis, I explored both regression and classification problems in machine learning, as well as supervised and unsupervised learning methods. In chapter 1, I explore a supervised, regression problem which attempts to estimate the parameter values involved in a Susceptible-Exposed-Infectious-Removed (SEIR) epidemiological model. Chapter 2 explores unsupervised learning where we use principal component analysis to find relationships in the genomic data of hosts and their viruses.

Chapter 1

A comparative study of likelihood-free methods for the estimation of epidemiological dynamics of respiratory viruses

1.1 Introduction

Prediction of infectious disease epidemics is essential to their control, but also a difficult process. This is because the epidemiological dynamics, i.e., the time evolution of the number of infected individuals, are nonlinear, with the probability of a susceptible individual acquiring infection depending on the number of infected individuals. Previous studies have constructed mathematical models describing the transmission dynamics of infectious disease,

known as the Susceptible-Infectious-Removed (SIR) model, and fit the model to the time series data of the number of infected individuals [4]. Conventional statistical methods, e.g., maximum likelihood estimation, require explicit solution of the time series data of the number of infected individuals from the SIR model. However, an explicit solution is difficult to obtain due to the nonlinearity of the model. Therefore several approximations are required to fit the SIR model with the epidemiological data of infectious diseases. Furthermore, the transmission of infectious disease is a stochastic event. A mathematical model taking into account stochasticity is required to estimate parameters.

One common property of transmission dynamics is the threshold for outbreak: an outbreak occurs only if the basic reproduction number, R_0, exceeds unity. In a biological sense, R_0 is the expected number of secondary infections by an infected individual when a population is fully susceptible [11]. Estimation of R_0 helps to predict the outbreak potential, final epidemic size, timing of the epidemic peak, and vaccination coverage required to prevent an outbreak. To estimate R_0, a common method is to fit the SIR model to epidemiological data. The simplest SIR model has only this one parameter, R_0, by scaling the unit time in the SIR model. In this paper, we use a Susceptible – Exposed – Infectious – Removed (SEIR) model, a variation of the SIR model. The SEIR model is comprised of additional parameters and follows more complex epidemiological dynamics, which reflect realistic disease dynamics.

Due to the importance of estimating R_0, numerous methods have been developed [31]. The accuracy of the estimates depends on both the estimation method and the data. For example, in one approach R_0 can be estimated from the slope of the time series data of infected individuals at the initial phase of an epidemic [33]. This method approximates the

epidemiological dynamics at the initial phase as an exponential growth. The accuracy of this method is sensitive to the period of epidemiological data available. An alternative approach estimates R_0 from the final epidemic size, i.e., the total number of infected individuals [46]. Because the relationship between the final epidemic size and R_0 cannot be described explicitly, the likelihood function of R_0 with an arbitrary final epidemic size cannot be described in an explicit form. Consequently numerical solutions or approximations are required to construct the likelihood function.

Recently a likelihood-free method has been proposed: approximate Bayesian computation (ABC) [40, 36]. This method approximates the posterior distribution using a rejection algorithm with the numerical integration of the SEIR model. This method is easy to implement, however several limitations remain. Some issues include a) parameter estimation takes a long time, particularly as the epidemiological model complexity increases and b) the ABC method accuracy is dependent on both the summary statistic and the accept/reject decision threshold, but there are no fixed rules for the selection of either.

A second likelihood-free approach has recently emerged in the form of machine learning (ML). The field of machine learning has grown rapidly with a large expansion of theories, applications, and algorithms. Problems can be categorized as either supervised or unsupervised and as classification or regression [3]. A supervised learning problem has a dataset and an answer, for example the pixels making up a photograph of a number can be a dataset and the numerical representation of the number is the answer (e.g., the number '7'). These two pieces of information are passed to the ML model during training so the model learns to recognize pixels of the type given as the answers it receives. Once the model is trained,

a separate, new dataset is given which contains only the pixel information. The model is then asked to predict the answer based on the data it had previously seen. This example of supervised learning is also an example of a classification problem. The number problem can be split into ten discrete categories (the whole numbers '0' to '9'), and the machine learns to classify the results into these categories. A regression problem, on the other hand, seeks to find a continuous value answer to the input it receives. Predicting housing prices is a common example of a regression problem, where, given a set of information about properties, the ML model can predict a continuous, numerical value estimate for the cost of the property.

Supervised ML models combine linear regression, gradient descent, maximum likelihood, and least squares functions to develop weight matrices and comparison functions to predict and estimate parameter outputs based on the historical knowledge of input/output pairs [3]. With the expansion of the field of ML, new models continue to be developed and improved, connecting the building blocks of ML in new ways to uncover hidden connections in data. Some methods, such as convolutional neural networks (CNN) are well suited for two-dimensional image analysis [25], while other methods, like long-short term memory models (LSTM), specialize in handling time series data [18].

In this study, we propose a ML approach to estimate the R_0 of a respiratory virus from a time series of incidences of the disease as a supervised regression problem. Additionally, we seek to estimate other parameters associated with the SEIR model and time series generation. As mentioned above, R_0 is highly dependent on the mathematical model. Our final goal is a likelihood-free estimation of R_0, as well as other model parameters. The ML methods used in this study are two separate multi-layer perceptrons (MLP), a CNN, and a LSTM model.

For reference and comparison, we also use the ABC approach to estimate the same values using the same datasets. We compare not only the accuracy of the two methods, with credible and confidence intervals, but also the time required by each approach to reach its answer. Of the four ML methods tested, the MLP with time model was the most robust as well as being significantly faster than the more complex CNN and LSTM models.

1.2 Materials and Methods

This study can be broken into five main parts. The first is the development of an individual-based (IBM) SEIR epidemiological model for generating data; the second is the ABC method used for estimating parameters; the third is the learning by MLP, CNN, and LSTM machine learning models, again to estimate parameters; the fourth is the dataset creation and bootstrapping of the real-world and test data to create confidence intervals on the machine learning solutions; and finally calculation of the time it took for each method to obtain its estimates.

The ML models were trained on 100000 datasets, validated on 1000 datasets, and tested on 1000 datasets. ABC was run on 1000 sample datasets and compared against a total of 100000 comparison datasets. The ABC sample and ML test datasets were the same and the ABC comparison and ML training datasets were the same. An explanation of these datasets is in the following section. We evaluate the accuracy of estimation by two measurements, the average error and the width of the credible interval for ABC and confidence intervals for ML. Each parameter range was divided into ten subranges. The errors among parameters

in each subrange were then averaged to create the 'average error'. The average width of credible/confidence intervals is the average difference between the lower and upper bound of the interval.

1.2.1 Terminology

As the different likelihood-free methods used in this paper (ABC and ML) each have their own standard vocabularies, we first clarify terminology in this paper to make a direct comparison of methods possible.

ML typically uses three datasets. The 'training' set is a large dataset which is given to the ML model during the learning phase. The 'test' set is a completely new and unseen dataset which the ML method passes through the trained model to estimate the posterior parameter values. The 'validation' or 'development' dataset, like the 'test' set is a new and unseen dataset used as an interim test. That is, this dataset is used to verify the model is learning, check accuracy of estimates, and when running trials of different hyperparameter sets. In ABC there is no dataset equivalent to ML's validation set.

According to the notation used by [40] we use the symbols D for the data in question, either real-world observed data or generated 'test' data, and $\hat{D}$ for comparison data. In ABC, $\hat{D}$ would normally be the Markov chain Monte Carlo generated data presented to the rejection algorithm. The rejection algorithm takes a set of generated data and compares it to the data in question, D. In ABC, the summary statistic is used to calculate the distance between $\hat{D}$ and D, and the parameters are accepted or rejected if the estimated distance falls beneath an accept/reject threshold. Our summary statistic for ABC is the Euclidean distance

between the dataset D and the simulated dataset $\hat{D}$:

$$\sqrt{\sum(D-\hat{D})^2}$$

Comparing the ML and ABC terminology, datasets comprising D would be the test set in ML, while a dataset comprising $\hat{D}$ is given to an ML algorithm as a training set. To compare the ABC and ML methods we use the same pre-generated datasets for both methods.

1.2.2 SEIR Epidemiological Model

The SEIR model is an expansion of the SIR model, which describes the time evolution of the number of infected individuals during a disease outbreak. The host population is classified by their health status, susceptible (S), exposed (E), infectious (I), and removed (R) (recovered or deceased). Transmission events happen via contact of S and E with constant rate β. The SEIR model can be expressed mathematically through the following simple equations [4, 31, 10]:

$$\begin{aligned}
N &= S(t)+E(t)+I(t)+R(t),\\
\frac{dS}{dt} &= -\beta S\frac{I}{N},\\
\frac{dE}{dt} &= \beta S\frac{I}{N}-\varepsilon E,\\
\frac{dI}{dt} &= \varepsilon E-\gamma I,\\
\frac{dR}{dt} &= \gamma I.
\end{aligned}$$

Here N describes the total host population size. In this model, R_0 is given by:

$$R_0 = \frac{\beta}{\gamma}.$$

This is a deterministic model and S, E, I and R are continuous. To fit the model with the data, it is required to expand this model to a stochastic one with discrete S, E, I and R. An individual-based SEIR model describes the stochastic process of transmission dynamics at the individual level. Let H_x be the health state of the x-th individual.

$$H_x \in \{S, E, I, R\}$$

The probability of transition between each health state can be written by:

$$Pr(H_x(t) = S \rightarrow H_x(t+\Delta t) = E) = \beta I(t)\Delta t,$$

$$Pr(H_x(t) = E \rightarrow H_x(t+\Delta t) = I) = \varepsilon \Delta t,$$

$$Pr(H_x(t) = I \rightarrow H_x(t+\Delta t) = R) = \gamma \Delta t.$$

We simulate this model to create data which can be used to learn the different parameter sets of R_0, ε, and γ. ε is the latent period, or the rate at which exposed individuals become infectious. γ is the recovery rate, the rate individuals move from state I to state R. The host population size for the general study, N=2225. For the real-world comparisons, N=500 for mumps [39], N=343 for measles [32], and N=2225 for influenza A(H1N1)pdm09 [27]. We

set $S(0) = N - I(0)$, $E(0) = 0$, $I(0) = 22$ and $R(0) = 0$ as the initial conditions, which were parameterized based on the epidemiological data of [27]. Time series data of incidence were created by IBM model with randomly chosen parameter sets to consist of 100000 training, 1000 validation, and 1000 test samples. Throughout this study we set Δt = 1/10 day, meaning the SEIR model parameters were updated ten times each day, with each change in time (Δt) equal to 1/10 day.

1.2.3 Approximate Bayesian Computation

We use ABC to estimate the parameters R_0, ε, and γ, the parameter values associated with the SEIR epidemiological model. For our ABC calculations, we used simulated datasets for $\hat{D}$, calculating the distances between each D dataset and $\hat{D}$ datasets. The Euclidean distance was used as the summary statistic to calculate distances between datasets, and multiple acceptance thresholds were established. An acceptance threshold of 60 was used for the figures 1.1, 1.3, 1.4, here as it was large enough to produce accepted posterior parameter sets for each of the D datasets and have enough samples to create credible intervals, but small enough to generally discriminate between similar and dissimilar datasets.

The Euclidean distance between the dataset D and simulated dataset $\hat{D}$ is:

$$\sqrt{\sum (D - \hat{D})^2}$$

1.2.4 Machine Learning

Three separate machine learning models and one variation were implemented and run with early-stopping manually executed by comparing the loss at each epoch, stopping when the loss no longer decreased (for 10 or 20 epochs), and using the weights from the best epoch, i.e. the epoch with the smallest loss. The learning rate for all models was 0.0001. The ML models in this study were implemented in Python using Lasagne and Theano [1] libraries. For all ML models, numerous hyperparameters for the number of hidden layers, number of hidden units, learning rate, activation functions, and number of training samples were explored and the hyperparameters which routinely yielded the best results were selected. For example, we ran an MLP model with 2, 3, 4, 5, 6, and 7 hidden layers and found that there was little benefit in models deeper than three hidden layers, yet two hidden layers learned poorly. For this reason, three hidden layers were used in the final model.

Multi-Layer Perceptron

The first model selected for this paper was a simple MLP where the entire time series dataset was passed in as a single input array and the parameters were estimated from learned relationships in that dataset. Small changes in parameter values have a large impact on the shape and behavior of the time series graph, so it was important for the MLP to see all of the data with equal importance, which is why the time series was passed in as a single input.

The MLP model accepted the time series information for the number of incidences of infection per day as created by the IBM model as input for learning. This input was given to the model as a single entity consisting of the number of newly infected individuals at each

timestep. It also received the 'answers' of the R_0, γ, and ε model parameters which were used to generate the time series data. The model was asked to learn these answer parameters in a supervised manner given the time series of incidences. The MLP model was created with three hidden layers having 400 hidden units per layer. The hidden layers used rectified linear units [30] for their activation functions, with a linear activation function in the final output layer.

Multi-Layer Perceptron with Time

A second MLP model was constructed to incorporate the concept of 'time' into the time series information. In a standard MLP, the sequence of values passed in has no connection and there is no concept of order in the analysis and training. This second model added a time element by creating a tuple consisting of the day and the number of new incidences that day. This simple method created a model which understands time as an individual value, though unlike LSTM described later, it does not have any kind of memory mechanism to compare previous and future datapoints.

This MLP also had three hidden layers, though with 400, 200, and 100 hidden units per layer, in that order. Again, the hidden layers used rectified linear units for their activation functions with a linear activation function in the final output layer. The addition of the time element changed the input to the model from a one-dimensional array to a two-dimensional matrix.

Convolutional Neural Network

While the CNN is not a traditional choice for problems involving time series data, it was selected in this case due to the complex nature of the SEIR modeling data. The mathematical models being simulated are highly non-linear, nearly chaotic at times. As CNNs are known to be capable of modeling very complex behavior [25], they were tested in this problem space for comparison.

The CNN model was constructed of two one-dimensional convolutional layers and two pooling layers, with a single dense hidden layer prior to the output. The convolutional and hidden layers all used rectified linear units for their activation functions, while the output layer again used a linear activation function.

Long-Short Term Memory

A recurrent neural network (RNN) approach, LSTM models are designed for analysis of time series data [18], like that found in this study. The memory aspect is built into the model using a memory cell and gates (input, output, and forget) to control the data in the model. These components ensure continuity in the data of the LSTM model and accept the time order as an important feature of the data.

The LSTM model implemented for this study consists of two LSTM layers with 16 hidden units and gradient clipping at 100 (to prevent exploding gradients). No activation function is applied in the LSTM layers, however a linear activation function is applied on the output layer.

1.2.5 Datasets

The datasets for this study come from two sources. The first, generated data, is a set of SEIR epidemiological model datasets generated using individual-based, Monte Carlo simulations, as described in the Epidemiological Model section. The second source comes from time series sets of incidences from published papers by [27], [39], and [32]. We estimate R_0 for these time series datasets and compare our estimates against the general or estimated value from the papers.

The data in these datasets are comprised of two parts. The first part is a time series of the number of newly infected individuals per day over the course of an outbreak. The second piece of information is the parameter set of R_0, γ, and ε used in the simulation to generate the time series. The time series is the information the ML model trains on, while the parameter set is the answer it is trying to achieve. In ABC, the time series is the dataset $\hat{D}$, and the parameter set is the answer it is estimating.

Bootstrap Resampling

To date, machine learning has most often been used in classification problems with discrete correct answers and myriad ways to determine the performance of any given model, such as accuracy, precision/recall, F-score, and receiver operator characteristic (ROC) [38]. Regression problems, by contrast, have few methods to explore the quality of the output. To address this issue, we created a novel method to build confidence intervals for the outputs of the ML model.

We started by creating a standard machine learning test set of 1000 datasets. Running this time series dataset through the trained model gives estimates for the parameter values, but there is no indication on the quality of the estimates - no certainty or confidence associated with the values. Next, we created 1000 bootstrap-resampled datasets for each test dataset, for a total of one million tests. For bootstrap resampling of the time series data of incidence, the time series data of incidence can be interpreted as the set of emergence times. For example, the data when incidence at $t = 1$ is 1 and incidence at $t = 2$ is 2 is equivalent to a set of emergence times, $\{t = 1, t = 2, t = 2\}$. We resampled the emergence times by bootstrap resampling from this set of times, and converted them back into time series data of incidence.

For the estimates returned for the 1000 resamples of each dataset, we calculated the mean, median, mode, and 95% confidence intervals. This method provides a measure of credibility for each estimated output parameter.

1.2.6 Time Calculations

The ABC computations were conducted on a server with 2.80 GHz processors and 1 TB of memory. The computations were run across multiple CPUs and the time calculated is the combined time to run all scripts. The ML computations were conducted on a server with 3.50 GHz processors, 64 GB memory, and a GeForce GTX 1080 Ti ® graphics card for GPU processing. Typically between four and eight processes were run simultaneously on the graphics card.

The time to complete each method is shown in table 1.1. For ABC, the time to verify its results by comparison of the distances between 1000 D datasets with 100000 $\hat{D}$ datasets

was measured, including time to split by thresholds and create credible intervals for accepted datasets. The time for a single comparison against 100000 $\hat{D}$ datasets was then measured for the test comparison. For the ML models, the time included the bootstrapping of the test dataset, training of the learning model, and computation of the confidence intervals. The test was then run by obtaining the estimates for a single bootstrapped sample, that is, running 1000 samples from one set of parameters through a trained ML model and collating the results.

1.2.7 Experiments on Real-World Datasets

For experiments, we used an SEIR model dataset with three parameters: R_0, γ, ε, calculating distances with ABC and comparing at various acceptance thresholds, and training CNN, LSTM, and two MLP models, then running a test dataset through the trained model. Finally, we tested our trained and verified models against three real-world datasets: 1) an outbreak of mumps [39], 2) an outbreak of measles [32], and 3) an outbreak of influenza A(H1N1)pdm09 [27] to see if we could accurately estimate the parameter R_0 with both our ABC and ML models.

For ABC, the time series of infectives is set as D and compared against the $\hat{D}$ generated data used throughout this paper. The required accept/reject threshold was 300, 20, and 20 for the influenza, measles, and mumps datasets, respectively. The accept/reject thresholds were selected where there were enough accepted datasets for the calculation of credible intervals. The parameters were then estimated from the accepted datasets. To construct confidence intervals of the estimates by ML, the real-world time series of incidence was resampled

using the bootstrap resampling method discussed in the Bootstrap Resampling section. The set containing the original and bootstrap resampled time series are given to the model, the parameters are estimated, and finally the confidence intervals are calculated from the ML model outputs.

1.3 Results

1.3.1 Comparisons of Average Errors

Figure 1.1 shows the average errors compared to the actual parameter values associated with each method. The average errors of estimates made with ABC and ML are most similar for R_0. ABC and MLP with time show nearly identical patterns, consisting of low errors when R_0 is less than 3.0 and then increasing with increasing values of R_0. ABC has slightly lower average error than MLP with time for all values of R_0. For R_0 below 2.25, MLP performs as well as ABC and MLP with time. For R_0 greater than 2.25 the average error from MLP estimates increases greatly, until R_0 reaches 5.0, at which point the error from MLP is approximately twice the error from ABC. CNN has a nearly constant average error of 0.2 for R_0 less than 3.8. For R_0 greater than 3.8, the average error on CNN estimates increases linearly to approximately 0.7 at $R_0 = 5.0$. Finally, LSTM shows erratic behavior in R_0 estimation. Until R_0 reaches 3.0, LSTM has the worst estimates among the methods tested, with an error reaching nearly 0.6. From 2.8 to 3.8, the average error by LSTM decreases, becoming smaller than all but CNN. From $R_0 = 3.8$ to 5.0, the error by LSTM increases again, with behavior similar to ABC, CNN, and MLP with time for this range of R_0.

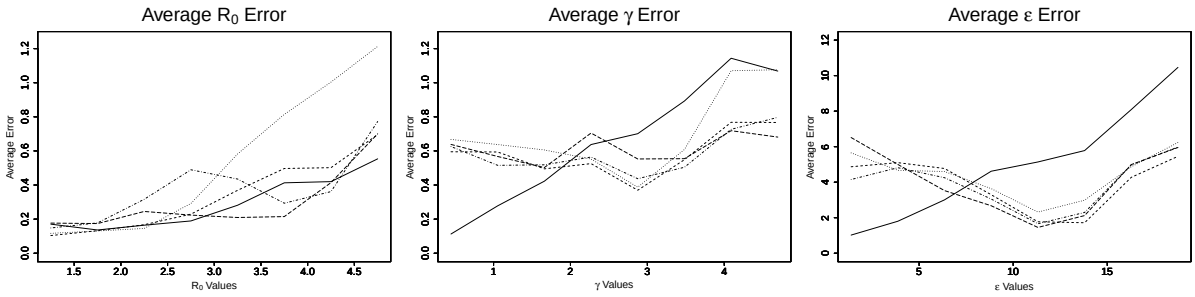

Fig. 1.1 Average error for ABC (solid line), CNN (long dashed line), MLP (dotted line), MLP with time (dashed line), and LSTM (dashed and dotted) estimates against actual parameter values for an SEIR model with three parameters, on 100,000 training datasets.

The patterns for average error for the parameters γ and ε are similar to one another. ABC follows one pattern while the ML solutions follow a different pattern. For ABC, the average error increases with increasing values of the parameters, γ and ε. For γ less than 2.0 $days^{-1}$ and ε less than about 7.5 $days^{-1}$, the ABC average error is smaller than the average error for all ML estimates. For γ more than 2.0 $days^{-1}$ and ε more than 7.5 $days^{-1}$, the average error for ABC is larger than the average error of all ML estimates. When the number of training samples is increased to one million, the point at which ML becomes more accurate than ABC is 1.6 $days^{-1}$ for γ and approximately 6.5 $days^{-1}$ for ε (see Supplemental Data). The ML approaches maintained a nearly consistent average error for all γ of 0.6 $days^{-1}$ until $\gamma = 3.9$ $days^{-1}$, when they increased to average errors of approximately 0.6 $days^{-1}$. From $\gamma = 3.9$ to 5.0 $days^{-1}$, MLP increased much more quickly to approximately 1.1 $days^{-1}$, the same as ABC. All four ML models behaved the same for ε, with average errors decreasing with increasing ε until $\varepsilon = 11$ $days^{-1}$, at which point they increased with increasing ε at a rate similar to ABC, but approximately 3 $days^{-1}$ smaller.

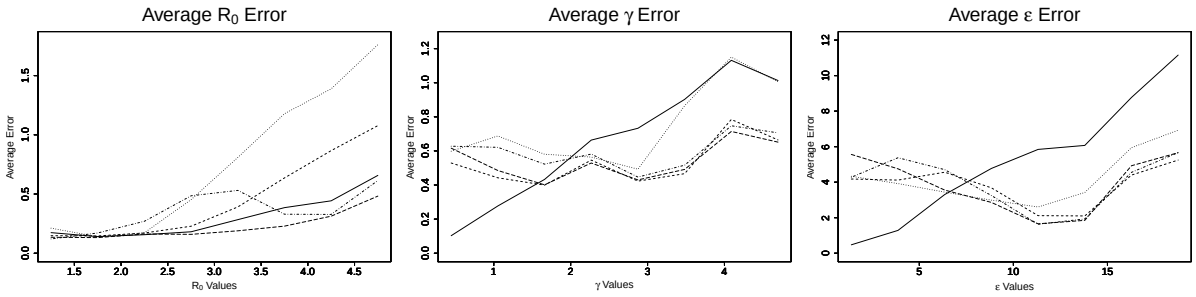

Fig. 1.2 Average error for ABC (solid line), CNN (long dashed line), MLP (dotted line), MLP with time (dashed line), and LSTM (dashed and dotted) estimates against actual parameter values for an SEIR model with three parameters, for one million training datasets.

1.3.2 Comparisons of Credible / Confidence Intervals

The credible intervals for the estimates for ABC and confidence intervals for ML are shown in figure 1.3. CNN has a constant size confidence interval for R_0. MLP and MLP with time have a confidence interval similar in shape to ABC' s credible intervals for R_0, though their confidence intervals are larger. These three methods start with small credible/confidence intervals for small R_0, increasing with R_0 until R_0 = 2.6 to 4.0 and then decreasing slightly. LSTM' s confidence interval increases until R_0 reaches approximately 2.25, then decreases with increasing R_0. The credible/confidence intervals for both the ABC and ML models decrease as γ increases. CNN has the smallest confidence interval for γ, followed by LSTM, MLP with time, ABC, and MLP. In ABC, the credible intervals for ε remain mostly constant at about 16.0 day^{-1} for all values of ε. For the ML models, the confidence intervals decrease with increasing ε, increasing slightly around ε = 15.0 $days^{-1}$ and then decreasing again. Again, CNN has the smallest confidence interval for ε, followed by LSTM, MLP with time, MLP (for ε greater than 5 $days^{-1}$), and ABC.

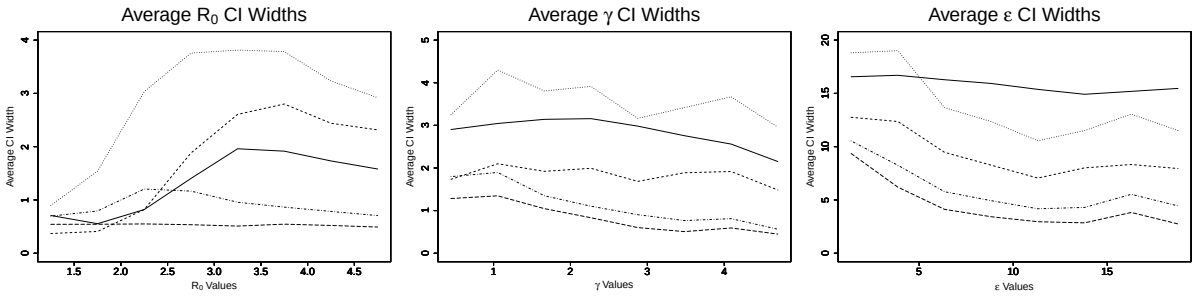

Fig. 1.3 Average credible/confidence interval width for ABC (solid line), CNN (long dashed line), MLP (dotted line), MLP with time (dashed line), and LSTM (dashed and dotted) estimates against actual parameter values for an SEIR model with three parameters, on 100,000 training datasets.

1.3.3 Comparisons of Estimated and Actual Values

Figure 1.4 shows the estimated parameter values compared to the actual values for each method. ABC estimates agree closely with the actual values for R_0 less than 3.0, with increasing error as R_0 grows beyond 3.0. MLP also estimates R_0 close to the actual values for R_0 less than 3.0. After R_0 = 3.0, however, the estimate by MLP is nearly constant at a little less than 3.0. CNN estimates R_0 close to the actual values while R_0 is less than 4.0. MLP with time' s trend is similar to ABC, with estimates near actual values for R_0 less than 3.0. LSTM shows close estimation for R_0 less than 2.0, then overestimates from 2.0 to 4.0, then underestimates for R_0 greater than 4.0. Overall, the ML methods appear to underestimate R_0 values, whereas ABC shows equal over- and underestimates. The ML methods show larger over- and underestimation than ABC for low R_0.

All ML methods generally overestimate all values for γ. LSTM, MLP with time, and CNN estimates were close for all γ, while MLP again began estimating a constant, approximately 3.0 $days^{-1}$ for γ greater than 3.0 $days^{-1}$. ABC closely estimates γ values less than 2.0 $days^{-1}$ and ε values less than 5.0 $days^{-1}$. ABC, and to a lesser extent MLP with time and

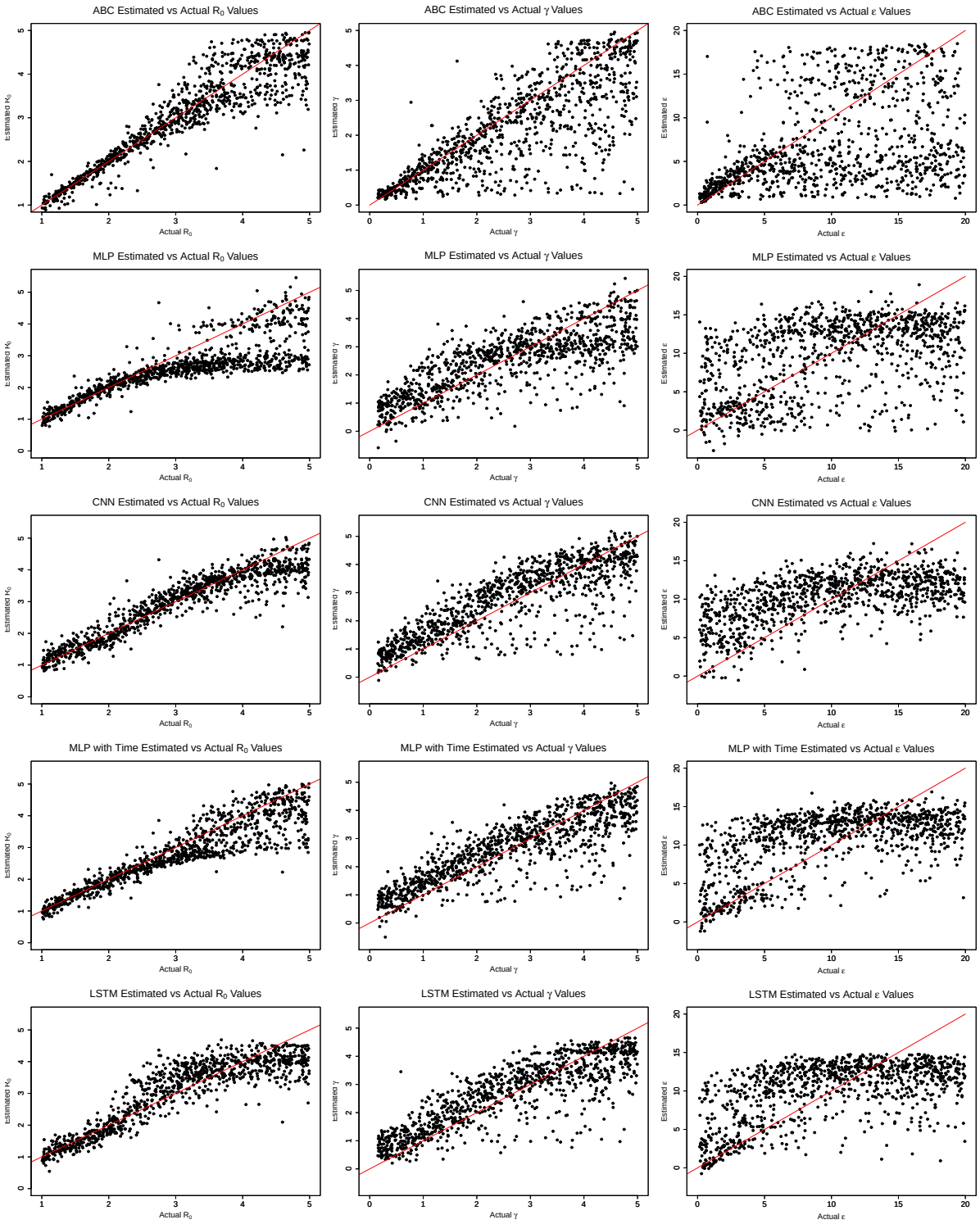

Fig. 1.4 The estimated and actual parameter values from ABC, MLP, CNN, MLP with time, and LSTM.

LSTM, appear to estimate ε somewhat closely for ε less than 5.0 $days^{-1}$, but for values greater than 5.0 $days^{-1}$ the results for all methods are nearly random.

1.3.4 Run Times

The time to complete each method is shown in table 1.1. The ABC method took 410 minutes to calculate the distances and accept/reject 1000 test datasets from 100000 training datasets. The time for a single dataset to be compared against 100000 training datasets was 0.65 minutes. The fastest ML model, MLP, took 68 minutes to train while the slowest, LSTM, took 364 minutes. MLP with time and CNN took 71 and 184 minutes to train, respectively. This makes training of the ML models between 1.1 and 6.0 times faster than the time to verify a similar ABC method on 100000 training datasets. Testing on a single dataset for the ML models took 0.03, 0.03, 0.05, and 0.06 minutes for MLP, MLP with time, CNN, and LSTM, respectively. These times are between 10.8 and 21.7 times faster than the estimate calculation for a single sample via ABC. When the size of the training set is increased from 100000 to one million, the time required to estimate the parameters for a single ABC test set scales linearly with the number of comparisons, however the estimation via machine learning remains constant regardless of the number of samples used to train the data (table 1.2).

Table 1.1 The time required to train 100,000 samples and test on a single sample for the ABC, MLP, CNN, and LSTM methods.

Method	Train	Test
ABC	410 min	0.65 min
MLP	68 min	0.03 min
MLP with time	71 min	0.03 min
CNN	184 min	0.05 min
LSTM	364min	0.06 min

Table 1.2 The time required to train one million samples and test on a single sample for the ABC, MLP, CNN, and LSTM methods.

Method	Train	Test
ABC	3962 min	4.75 min
MLP	420 min	0.03 min
MLP with time	531 min	0.03 min
CNN	747 min	0.05 min
LSTM	784 min	0.06 min

1.3.5 Application to Real-World Epidemiological Data

We also compared ABC and ML with epidemiological data for mumps, measles and influenza.

R_0 for mumps has been estimated between 3.6 and 4.5 [15]. Table 1.3 shows the comparison of ML and ABC estimations of R_0 with the typical real-world values. MLP, MLP with time, and CNN estimated R_0 at approximately 4.0 using the data from [39] for an outbreak in Centerville, OH, USA. ABC estimated R_0 as slightly lower at 3.74, while LSTM greatly underestimated R_0 at 2.71. These values were created based on an estimation of the effective reproductive number and the vaccine coverage of students within the school of 72.7%.

Table 1.3 The estimation results for R_0 from previous published research on mumps and ABC, two MLP, CNN, and LSTM models. Estimates shown include the 95% credible/confidence intervals.

Methods	R_0 for Mumps
Previous study	3.6-4.5 (Edmunds [15])
ABC	3.74 (1.21, 12.09)
MLP	4.07 (1.10, 9.08)
MLP with time	3.92 (3.41, 4.25)
CNN	4.21 (2.53, 4.76)
LSTM	2.71 (1.83, 4.54)

For an outbreak of measles at Wincrange, Luxembourg [32], the effective reproductive number was estimated as 1.5 (95% CI: 0.9, 2.2). Table 1.4 shows the comparison of ML

and ABC estimations with the effective reproductive number estimated for the outbreak. All five of our methods underestimated the effective reproductive number of measles, but their estimates fell within the 95% CI estimated by the previous study.

Table 1.4 The estimation results for effective reproduction number from previous published research on measles and ABC, two MLP, CNN, and LSTM models. Estimates shown include the 95% credible/confidence intervals.

Methods	Effective Reproduction Number of Measles
Previous study	1.5 (Mossong [32])
ABC	1.16 (0.59, 4.88)
MLP	1.00 (0.63, 1.88)
MLP with time	0.91 (0.69, 1.07)
CNN	0.96 (0.19, 1.26)
LSTM	1.08 (0.80, 1.52)

The estimated R_0 of an outbreak of influenza at a high school in New York during the 2009 influenza pandemic was 1.23 [27]. Table 1.5 shows the comparison of ML and ABC estimations of R_0 with the effective reproductive number estimated for the outbreak. ABC estimated R_0 nearly exactly at 1.24. MLP with time slightly underestimated the value at 1.06 (95% CI: 0.86, 1.30). Standard MLP greatly underestimated R_0 at 0.40, while CNN and LSTM greatly overestimated the R_0 at 2.84 and 1.84, respectively.

Table 1.5 The estimation results for R_0 from previous published research on influenza and ABC, two MLP, CNN, and LSTM models. Estimates shown include the 95% credible/confidence intervals.

Methods	R_0 for Influenza
Previous study	1.23 (Lessler [27])
ABC	1.24 (0.96, 1.74)
MLP	0.40 (0.00, 0.79)
MLP with time	1.06 (0.86, 1.30)
CNN	2.84 (2.08, 3.35)
LSTM	1.84 (1.38, 3.54)

Based on the general analysis for various R_0 values shown in figures 1 to 3, most of the results above agree. For the mumps outbreak, the effective reproductive numbers were estimated around 1.0 to 1.2. At this low range of R_0, based on Figure 1, the average error for all methods is relatively low, though LSTM and ABC have higher average errors than the other methods and also exhibit a small tendency to underestimate R_0 for values near 1.0. All five methods also underestimated the effective reproduction number for measles, which was estimated at 1.5. With the exception of MLP with time, the methods contained 1.5 in their 95% CI intervals. However, the estimates for all five methods were within the 95% CI of the previous study. Finally, for the flu estimates, CNN and LSTM greatly overestimate R_0 as 2.84 and 1.84, respectively. LSTM both over- and underestimates values around R_0 = 1.2. The estimation of 2.84 by CNN is not robust with expected values, as though CNN does tend to overestimate more than underestimate at R_0 = 1.2, an R_0 of 2.84 is outside its average error window.

1.4 Discussion

In this study we applied ML methods to estimate the epidemiological parameters of infectious diseases, and compared their accuracy and speed with ABC. In general, the width of confidence intervals estimated by ML are smaller than the credible intervals estimated by ABC. The average error of ML estimates are similar to ABC for R_0, and larger for small values, but smaller for large values of γ and ε. Furthermore, the ML models were faster to train than ABC.

ABC was more robust to changes in the data, as shown in tables 1.3, 1.4, and 1.5. MLP with time was the most robust of the ML methods, with a tendency to underestimate R_0. Given the difference in calculation times between ABC and MLP with time (410 minutes for ABC compared to 71 minutes for MLP to train 100000 samples and 3962 minutes for ABC compared to 531 minutes for MLP on one million samples), it is worth exploring methods which can reduce the underestimation of R_0 in the MLP with time solution. Possibilities include increasing the amount of training data, hyperparameter tuning, increasing the depth of the model, and other general ML tuning methods which may be applied [3]. The ML methods estimated γ well, but with an obvious overestimation bias which can be observed in Figure 1.4, which may also be corrected by applying the previously mentioned approaches.

Interestingly, the point at which ABC is better than ML shifts with increasing sample size for γ and ε. When trained on one million datasets, this point decreased from less than 2.0 to 1.6 $days^{-1}$ for γ and from less than 7.5 to 6.5 $days^{-1}$ for ε. CNN and MLP with time showed similar estimation capability with ABC for R_0, with average errors around 0.2 for R_0 less than 3.0 and increasing with increasing R_0. The MLP and LSTM approaches showed poor estimation ability for R_0.

While the ML models were faster to train than ABC was to verify, it should be noted that ABC verification of varying parameter values is not required, but ML training on all parameter values is necessary [34]. That is, an ABC test estimate can be made in approximately less than one minute without thorough verification of the general efficacy of the method, while the ML models must be fully trained before calculating a test estimate. However, once trained,

the ML models do not need to be retrained unless there is a large amount of new data or some other reason arises to retrain the model [3].

The problem of parameter estimation explored in this paper can be classified as a ML regression problem where the values of the estimates are continuous. The vast majority of ML research is on classification problems with discrete solutions and therefore 'right' and 'wrong' answers [13]. In bioinformatics and medicine, another characteristic of a large amount of ML solutions is the use of 2-dimensional images, again typically for classification, for example identifying breast cancer or analyzing MRIs [25, 7, 35]. One example of ML being used for regression comes from the European Space Agency (ESA) [44, 5], where neural networks and regression methods were explored for use in analyzing the large quantity of data being returned by the Sentinal-2 and Sentinel-3 satellites searching for life on far planets. Overall, the application of ML methods on regression problems requires further analysis to improve accuracy.

Note that the R_0 value of 1.2 cited from the paper by [27] is the estimate made over the entire course of the outbreak. This value agrees with existing genetic analysis of the virus, as well as additional epidemiological studies which estimated the R_0 value of influenza A(H1N1)pdm09 between 1.4 and 1.6 [16].

Several disadvantages of ML for estimation of epidemiological parameters were found. First, ML approaches are highly sensitive to the size of parameter ranges [29]. As parameter ranges increase, accuracy of the estimates decreases. In additional tests, we used normalization and standardization to try to reduce the impact of range size on model estimatibility, with limited success [19]. Moreover, ML is not robust to changes in the initial condition of

the model [24], even outside the parameters of interest, though ABC, too, showed sensitivity to initial conditions. This sensitivity may be reducible with larger datasets, deeper models, or the introduction of pruning algorithms and may be explored in later papers [2].

In this paper, we have explored a single set of continuous time data, capturing parameters as constant values in a mathematical model. The transmission process of infectious disease does not strictly follow mathematical models and parameters can change values over time. Two approaches for future work which would partially address these issues are 1) a "discrete time analysis" to observe changes in parameter values over time and 2) testing the robustness of our estimates by checking values from different epidemiological models. "Discrete time analysis" is a discretization of the time component of our model with the assumption that the parameter values are constant between time intervals to observe changes in parameter values over time. To check the robustness of our estimates, but still using simulated data, we could create multiple datasets from the Susceptible-Exposed-Infectious-Removed-Susceptible (SEIRS) [9] epidemiological model and use this much more complex data to test a model trained from simpler SIR or SEIR model data. This would check the robustness of the systems to changes in unknown parameters and allow us to observe and estimate the sensitivity of our systems. Furthermore, using an SEIRS data model for data generation would allow for analysis of longer and recurring epidemics [9] and the efficacy of ML and ABC in estimating more complex disease dynamics.

In conclusion, we have confirmed that both ABC and ML can estimate SEIR model parameters, with ABC and MLP with time being the most robust methods for different SEIR models and parameters. ML models learn more quickly than ABC can be verified,

however ABC verification is highly parallelizable, i.e. the problem can be broken into several processes and estimated concurrently, while the learning time for ML models is more difficult to reduce. A key benefit of ML is the speed with which new datasets can be analyzed. A single, new sample can be analyzed in a few seconds, compared to several minutes by ABC, and is constant regardless of the number of datasets used for training. This means a trained ML model would be helpful when estimating large batches of new data.

Chapter 2

The Search for Host-Specific Signatures in Viral Genomes

2.1 Introduction

In recent years, between 60% and 75% of emerging and re-emerging diseases worldwide have been zoonotic in nature [45]. Zoonoses are diseases which transfer from animals to humans and include both viruses and bacteria. Examples of viral zoonoses are influenza, Ebola virus disease, West Nile virus disease, Severe Acute Respiratory Syndrome (SARS), and Middle East Respiratory Syndrome (MERS). In recent years, these diseases have had a large negative impact on human health and society.

A key aspect to understanding and controlling zoonotic disease is identification of the disease' s natural reservoir host. Having an understanding of which species carry viruses, we can establish appropriate care and contact procedures, ensure separation between certain

animals and humans, or otherwise control human-to-animal contact. When the species of origin for a disease is unknown, as is the case with the recent Ebola virus outbreak in western Africa, we cannot pinpoint the outbreak location or take steps to prevent re-infection from the origin species. This prevents us from actively eliminating likely sources of infection and re-emergence of the disease.

Existing research suggests that viruses have a mechanism which allows them to adapt their genomes to that of their hosts [17, 14]. One observed example is the basic A+U (i.e. the number of A or U bases) content analysis of the H3N2 influenza A virus in humans. Earlier analysis by this group looked at the percentage of A and U bases in a genome and identified changing patterns over time. Since its most recent re-introduction to humans was in 1968, the A+U content in the human H3N2 viral genome has been decreasing, becoming more similar to the genomic content of humans than its original (reservoir) host, the wild duck. Through our research, we have re-confirmed this trend in nucleotide composition change and also that the same trend is not observed in the genomes of viruses which have remained in the duck species. That is, it is a phenomenon observed only in the non-reservoir host, indicating an adaptation of the virus to humans.

The existence of this relatively easily observable viral adaptation provides us with evidence that it may be possible to identify the specific host species from the virus genomes alone by comparing the content patterns of the viral genome to the content patterns of a variety of potential host species. Increased similarity between virus and potential host genomes is expected to suggest closer relationship-potential and possibly identification of a single host species or family.

The purpose of this research is to clarify the relationships between host and viral genomes, as related to the adaption of a virus to ensure its continued survival in a given host. To achieve this goal, we aim to develop a system which, given a viral genome, can detect the original viral host. Some keys to viral host identification include multiple host and viral genome content analysis (k-mer analysis, including single, di-, tri-, and etc. nucleotide composition analysis), the mapping of key similarities and differences between reservoir hosts, non-reservoir hosts, and the viruses, and changes in the viral genome in a host over time. By first obtaining these intermediary results and identifying signatures unique to a virus and reservoir host relationship, a system which can infer the original viral host may be developed. Once the original or reservoir host for a virus is identified, treatment and prevention of diseases caused by these viruses can be improved, and transmission to humans can be reduced, even eliminated. However, for many viruses, these benefits cannot be seen until we have definitively identified the virus' origin.

As noted by Duffy, et al. (2008) [14], single-stranded RNA viruses tend to be smaller, have higher replication speed, and be more likely to be transmitted directly than double-stranded DNA viruses. These factors affect the mutation and substitution rates in viruses, making single-stranded RNA viruses more likely to contain more easily distinguished host-identifying modifications. For this reason and the fact that previous research [17, 42, 20, 28, 8, 22] has focused more on single-stranded RNA viruses, this current research also looks at single-stranded RNA viruses initially, with a goal of moving to larger, more complex viruses over time.

Fruit bats and wild ducks were selected for this research for two separate reasons. First, as wild ducks are the accepted reservoir host for the influenza A virus [21], this seemed an obvious first choice, as the influenza A virus can be transmitted to both humans and swine making it a good species for analysis purposes. Additionally, any analysis conducted must find and accept this previously verified connection to be considered valid, making the relationships between the host wild duck and influenza A virus a good validation test for any future approaches. The fruit bat was selected as it is a suspected carrier of numerous zoonotic diseases, though some viruses, such as the Zaire Ebola virus, have not been proven to have originated from the fruit bat [6, 26]. Evidence also exists suggesting that bats may not be the natural reservoir and this research, upon completion, could help confirm or exonerate fruit bats as the natural hosts of several viral infections.

2.2 Materials and Methods

The nucleotide sequences used in this research were obtained from the National Center for Biotechnology Information (NCBI) databases, including the NCBI GenBank and Influenza Database, and were obtained via either direct download or scripted data pulls. The full genomes for the fruit bat and mallard duck were downloaded by accession number from the NCBI databases. A selection of single-stranded RNA viruses for both wild ducks and bats were downloaded. A full list of accession numbers has been included in the Accession Number section.

Once obtained, the sequences were screened for bad data (i.e. any character which was not a nucleotide was removed), converted to single-line reads (all new-lines, carriage returns, etc. were removed from the reads), and joined into a single file. Table 2.1 shows the numbers of sequences examined for each group and the total number of nucleotides used.

Table 2.1 The nucleotide sequences of bat, bat virus, duck and duck viruses used in this study.

Gene Set	Number of nucleotide sequences	Total number of nucleotides (bp)
Bat genes	703	1,902,985,556
Bat virus genes	45	220,453
Duck genes	154	101,898,546
Duck virus genes	27	242,847

Next, the organized data was analyzed for single and dinucleotide content, including pure numerical counts, percentage of each single and di- nucleotide, percentage of A-or-U and C-or-G, and a ratio for each dinucleotide. This ratio was calculated as the quantity of the dinucleotide observed in the read divided by the product of the percentage of each of the individual nucleotides.

These values, single and di- nucleotide percentages, overall A+U and G+C percentages, and ratios, were then used in a principal component analysis (PCA) in R. The PCA was run and graphed against different combinations of the data, for example all duck data graphed against all bat data, virus data grouped against host data, host against host, and virus (by host) against virus.

This method can be used on any species or family of hosts, but as the original analysis by Greenbaum, et al. [17] showed effectiveness solely with single-stranded RNA viruses, these calculations were also conducted on single-stranded RNA viruses only. Once signatures have

been identified, we can broaden the scope of viruses under review to investigate whether the signatures are universal or restricted to a subset of viruses.

2.3 Results

Table 2.2 shows the frequency of each dinucleotide in the genomes of the bat and duck as well as the overall frequency of each dinucleotide in the set of single-stranded RNA viruses for each host species.

The first principal component, shown in Figure 2.1, is the overall G+C content of the genome. The following principal components are most closely related to the rate ratio of TpC (T preceding C), the observed and rate ratio of GpA, and the observed GpT dinucleotides.

Table 2.3 shows the percentage of the variance in the data that can be described by each principal component. While it takes 14 components to explain 99% of the variance, the first six principal components explain nearly 80% of the variance. A total of 38 components were analyzed, though not all uniquely described the data (for example, both the A+U and G+C percentages were included as components, and combined describe the x-axis of the PCA analysis (i.e. the first principal component)).

2.4 Discussion

Previous research has used the %G+C and CpG to define distinction between viruses and their hosts [17, 42, 20, 28, 8, 22]. However, the PCA results from this research suggest that CpG is not a good component for identifying the host species of a given virus.

Table 2.2 Frequency of dinucleotides in bat, bat viruses, duck, and duck viruses.

Dinucleotide		Bat genes	Bat virus genes	Duck genes	Duck virus genes
AA	Frequency	10.1%	8.5%	9.6%	10.1%
	Rate Ratio	1.12	1.05	1.16	1.01
AC	Frequency	5.0%	5.9%	5.2%	5.9%
	Rate Ratio	0.84	1.04	0.85	0.91
AG	Frequency	6.9%	6.3%	7.3%	7.8%
	Rate Ratio	1.15	1.01	1.19	1.03
AT	Frequency	8.1%	7.7%	6.8%	7.8%
	Rate Ratio	0.89	0.91	0.82	1.03
CA	Frequency	6.9%	6.9%	7.6%	8.2%
	Rate Ratio	1.15	0.78	1.25	1.26
CC	Frequency	4.9%	4.1%	5.1%	4.5%
	Rate Ratio	1.24	1.05	1.12	1.07
CG	Frequency	1.2%	2.3%	1.3%	2.2%
	Rate Ratio	0.31	0.51	0.28	0.45
CT	Frequency	6.9%	6.5%	7.3%	5.7%
	Rate Ratio	1.15	1.11	1.19	1.15
GA	Frequency	6.0%	6.4%	5.8%	8.7%
	Rate Ratio	1.01	1.02	0.95	1.16
GC	Frequency	3.9%	4.4%	5.2%	4.5%
	Rate Ratio	1.00	1.01	1.15	0.92
GG	Frequency	4.9%	4.9%	5.1%	6.2%
	Rate Ratio	1.24	1.01	1.13	1.09
GT	Frequency	5.0%	6.3%	5.1%	4.4%
	Rate Ratio	0.84	0.97	0.85	0.77
TA	Frequency	7.1%	6.6%	5.8%	4.6%
	Rate Ratio	0.78	0.78	0.70	0.61
TC	Frequency	6.0%	5.4%	5.8%	5.6%
	Rate Ratio	1.01	0.92	0.95	1.14
TG	Frequency	6.9%	8.5%	7.6%	7.6%
	Rate Ratio	1.15	1.30	1.25	1.33
TT	Frequency	10.2%	9.1%	9.5%	6.1%
	Rate Ratio	1.12	1.03	1.16	1.06

Fig. 2.1 The principal component analysis has been conducted on all of the bat and duck genes and viruses, and the results displayed separated into the 'Duck' group (dark blue) and the 'Bat' group (light blue), with ellipses around the corresponding groups.

As shown in Figure 2.1, the duck and bat data overlaps by a large degree, however there is also some group separation based on the ellipses. This suggests alternative principal components may be successful in identifying host-specific signatures within viral genomes. Continuing with principal component analysis and expanding into discriminant analysis and additional host and viral species has the potential to uncover unique signatures in viral genomes. The separation of the ellipses, though incomplete, suggests that by maximizing the

Table 2.3 Percentages of variance explained by the principal components over all reads (bat, duck, and all bat and duck viruses).

Principal Components	Percentage of variance (%)	Cumulative Percentage of variance (%)
PC1	38.4	38.4
PC2	11.0	39.4
PC3	9.2	58.6
PC4	8.1	66.7
PC5	7.0	73.7
PC6	5.9	79.7
PC7	4.3	84.0
PC8	3.7	87.7
PC9	2.9	90.6
PC10	2.6	93.2
PC11	2.5	95.8
PC12	1.6	97.4
PC13	1.4	98.7
PC14	0.2	99.0

differences between the bat and duck data using discriminant analysis, we may discover the signatures we are looking for.

Previous research has typically chosen the ratio of observed CpG dinucleotides to the expected value following random distribution (described here as a 'rate ratio', but in other documents as an 'odds ratio') as a principal component when describing virus and host genetic similarities [17, 20, 28, 8, 22, 42]. Due to the unique nature of the CpG dinucleotide in vertebrate and other organisms, this has been a good point to begin analysis of virus and host similarities and differences. However, the technique has also led to some unexplained anomalies, such as the TpA dinucleotide, which does not follow the mathematical model predictions when the focus is on the CpG dinucleotide [41]. Using principal component analysis to automatically determine the components of primary interest, we may be able to produce a model in the future without the currently observed anomalies.

Several theoretical studies have been conducted to determine to what extent the machinery of host cells affect the characteristics of both hosts and pathogens. Karlin, et al. [23] constructed a mathematical model investigating codon frequencies and choices/biases in coding regions in the human genome. The research of Shackelton, et al. [37] focuses on large DNA viruses, while van Hemert, et al. [43] investigate large retroviral RNA viruses. Early research by Karlin, et al. [22] suggested that CpG content in small eukaryotic organisms was suppressed, but not in larger ones. Supposing we find host-identifying signatures, it would be interesting in future studies to see if those signatures hold with large DNA and more complicated RNA viruses. Additionally, if the CpG content is not suppressed in these larger viruses, finding the signature mechanism, or principal components, may also give us a better understanding of how these diseases change over time. Finally, we would like to investigate if codon usage biases within the host affect the viral genome and if they do, to uncover the characteristics of the affect. If we can discover a set of signatures, we may be able to answer some interesting questions on virus-host interactions and adaptation.

2.5 Conclusion

This research suggests that there may be new techniques that can be used to analyze the nucleotide composition of viruses and their hosts which may provide more accurate and robust models of similarities. Further research in this area may yield a set of 'signatures' between viruses and their hosts, allowing us to determine a virus' s host strictly from viral genomic information.

2.6 Accession Numbers

Bat Accession Numbers: NW_006494508.1, NW_006494469.1, NW_006492022.1, NW_006438879.1, NW_006442107.1, NW_006443131.1, NW_006439192.1, NW_006489824.1, NW_006440629.1, NW_006436284.1, NW_006441953.1, NW_006442110.1, NW_006492112.1, NW_006431044.1, NW_006442489.1, NW_006434844.1, NW_006494608.1, NW_006436285.1, NW_006491054.1, NW_006430115.1, NW_006429662.1, NW_006494078.1, NW_006441895.1, NW_006439191.1, NW_006439568.1, NW_006431382.1, NW_006434839.1, NW_006443124.1, NW_006431388.1, NW_006431912.1, NW_006472930.1, NW_006434645.1, NW_006440128.1, NW_006440150.1, NW_006494520.1, NW_006431911.1, NW_006435335.1, NW_006429308.1, NW_006488774.1, NW_006429654.1, NW_006431288.1, NW_006433224.1, NW_006480049.1, NW_006436811.1, NW_006429637.1, NW_006484720.1, NW_006440135.1, NW_006441061.1, NW_006434865.1, NW_006430503.1, NW_006494573.1, NW_006493778.1, NW_006482907.1, NW_006435702.1, NW_006491762.1, NW_006494554.1, NW_006432945.1, NW_006485340.1, NW_006481969.1, NW_006440149.1, NW_006440643.1, NC_026465.1, NW_006492782.1, NW_006443570.1, NW_006436278.1, NW_006432224.1, NW_006492344.1, NW_006439487.1, NW_006429600.1, NW_006494491.1, NW_006436830.1, NW_006438163.1, NW_006438311.1, NW_006494164.1, NW_006493879.1, NW_006431916.1, NW_006442487.1, NW_006436007.1, NW_006429596.1, NW_006433940.1, NW_006436757.1, NW_006439280.1, NW_006437767.1, NW_006481078.1, NW_006441105.1, NW_006435764.1, NW_006435763.1, NW_006437465.1, NW_006440137.1, NW_006432447.1, NW_006432669.1, NW_006439722.1, NW_006440644.1, NW_006441449.1, NW_006436696.1, NW_006434858.1, NW_006494030.1, NW_006433390.1, NW_006440624.1, NW_006441138.1, NW_006494148.1, NW_006432227.1, NW_006437392.1, NW_006430831.1,

NW_006440619.1, NW_006443225.1, NW_006441946.1, NW_006494516.1, NW_006491884.1, NW_006429864.1, NW_006441955.1, NW_006430045.1, NW_006434848.1, NW_006438995.1, NW_006441062.1, NW_006492911.1, NW_006441606.1, NW_006436047.1, NW_006437741.1, NW_006436295.1, NW_006434829.1, NW_006430147.1, NW_006489992.1, NW_006435066.1, NW_006431075.1, NW_006432449.1, NW_006492580.1, NW_006432595.1, NW_006488235.1, NW_006431923.1, NW_006435780.1, NW_006494408.1, NW_006436493.1, NW_006483737.1, NW_006494613.1, NW_006430723.1, NW_006436109.1, NW_006429598.1, NW_006484242.1, NW_006432938.1, NW_006488657.1, NW_006442597.1, NW_006492285.1, NW_006430057.1, NW_006436668.1, NW_006430472.1, NW_006435787.1, NW_006443566.1, NW_006436543.1, NW_006433727.1, NW_006438190.1, NW_006438073.1, NW_006431932.1, NW_006493911.1, NW_006438979.1, NW_006437748.1, NW_006431936.1, NW_006438074.1, NW_006439231.1, NW_006432468.1, NW_006438865.1, NW_006494544.1, NW_006434373.1, NW_006431922.1, NW_006438871.1, NW_006492089.1, NW_006436672.1, NW_006431077.1, NW_006494271.1, NW_006483702.1, NW_006494272.1, NW_006494479.1, NW_006433456.1, NW_006442508.1, NW_006433384.1, NW_006493560.1, NW_006434156.1, NW_006434897.1, NW_006432943.1, NW_006440951.1, NW_006431443.1, NW_006434704.1, NW_006477069.1, NW_006439239.1, NW_006439187.1, NW_006433377.1, NW_006433941.1, NW_006434828.1, NW_006464012.1, NW_006494323.1, NW_006440205.1, NW_006494159.1, NW_006431171.1, NW_006430061.1, NW_006434851.1, NW_006431909.1, NW_006431202.1, NW_006437801.1, NW_006442207.1, NW_006430048.1, NW_006441436.1, NW_006441420.1, NW_006481300.1, NW_006432681.1, NW_006442509.1, NW_006433072.1, NW_006480467.1, NW_006442113.1, NC_007393.1, NW_006441398.1, NW_006442117.1, NW_006484765.1, NW_006439247.1, NW_006439816.1,

NW_006438462.1, NW_006436742.1, NW_006442483.1, NW_006430734.1, NW_006441954.1, NW_006439205.1, NW_006493299.1, NW_006434441.1, NW_006492751.1, NW_006436291.1, NW_006491296.1, NW_006442105.1, NW_006493095.1, NW_006436283.1, NW_006443128.1, NW_006433133.1, NW_006437736.1, NW_006491741.1, NW_006494455.1, NW_006429475.1, NW_006430224.1, NW_006442112.1, NW_006431558.1, NW_006441387.1, NW_006440657.1, NW_006438096.1, NW_006431908.1, NW_006437855.1, NW_006494099.1, NW_006434574.1, NW_006441486.1, NW_006494203.1, NW_006439712.1, NW_006440136.1, NW_006430730.1, NW_006476801.1, NW_006436281.1, NW_006431915.1, NW_006491138.1, NW_006430461.1, NW_006436280.1, NW_006431495.1, NW_006489010.1, NW_006429595.1, NW_006430209.1, NW_006493176.1, NW_006434181.1, NW_006491626.1, NW_006430460.1, NW_006431831.1, NW_006438113.1, NW_006430724.1, NW_006437999.1, NW_006490348.1, NW_006432940.1, NW_006436286.1, NW_006429624.1, NW_006493742.1, NW_006440638.1, NW_006437367.1, NW_006487911.1, NW_006432527.1, NW_006438867.1, NW_006442345.1, NW_006443122.1, NW_006437738.1, NW_006436814.1, NW_006429350.1, NW_006440654.1, NW_006434719.1, NW_006441995.1, NW_006442122.1, NW_006440656.1, NW_006440002.1, NW_006489054.1, NW_006434097.1, NW_006436298.1, NW_006493370.1, NW_006443126.1, NW_006440830.1, NW_006440649.1, NW_006435781.1, NW_006430561.1, NW_006439249.1, NW_006429440.1, NW_006438906.1, NW_006430563.1, NW_006429686.1, NW_006434882.1, NW_006443138.1, NW_006439987.1, NW_006442830.1, NW_006441197.1, NW_006441066.1, NW_006443145.1, NW_006494260.1, NW_006443564.1, NW_006432372.1, NW_006437739.1, NW_006441399.1, NW_006430462.1, NW_006436810.1, NW_006434825.1, NW_006438090.1, NW_006486371.1, NW_006442490.1, NW_006434120.1, NW_006437924.1, NW_006429894.1, NW_006430173.1,

NW_006493047.1, NW_006493210.1, NW_006442514.1, NW_006429615.1, NW_006437816.1, NW_006432959.1, NW_006483048.1, NW_006438673.1, NC_002612.1, NW_006486721.1, NW_006435788.1, NW_006477324.1, NW_006488175.1, NW_006443218.1, NW_006434874.1, NW_006435777.1, NW_006492497.1, NW_006442592.1, NW_006488127.1, NW_006430534.1, NW_006430212.1, NW_006442847.1, NW_006441590.1, NW_006437746.1, NW_006429163.1, NW_006478250.1, NW_006437344.1, NW_006493875.1, NW_006437489.1, NW_006432235.1, NW_006442835.1, NW_006436671.1, NW_006432491.1, NW_006434709.1, NW_006436666.1, NW_006482297.1, NW_006440757.1, NW_006437456.1, NW_006486417.1, NW_006441950.1, NW_006491984.1, NW_006490036.1, NW_006493403.1, NW_006432497.1, NW_006493112.1, NW_006430738.1, NW_006435916.1, NW_006443204.1, NW_006436664.1, NW_006436289.1, NW_006432462.1, NW_006435766.1, NW_006435824.1, NW_006490817.1, NW_006438069.1, NW_006440132.1, NW_006430735.1, NW_006431525.1, NW_006431805.1, NW_006440676.1, NW_006443129.1, NW_006434863.1, NW_006431924.1, NW_006442493.1, NW_006447515.1, NW_006438082.1, NW_006491740.1, NW_006437753.1, NW_006438643.1, NW_006429112.1, NW_006439718.1, NW_006491357.1, NW_006438072.1, NW_006438869.1, NW_006442145.1, NW_006429634.1, NW_006434834.1, NW_006440655.1, NW_006489004.1, NW_006442158.1, NW_006433797.1, NW_006437788.1, NW_006440838.1, NW_006494290.1, NW_006430608.1, NW_006430525.1, NW_006433729.1, NW_006435793.1, NW_006434366.1, NW_006434901.1, NW_006429782.1, NW_006442484.1, NW_006490505.1, NW_006435778.1, NW_006434002.1, NW_006435758.1, NW_006431920.1, NW_006440642.1, NW_006431917.1, NW_006433381.1, NW_006432391.1, NW_006494019.1, NW_006494095.1, NW_006441943.1, NW_006431048.1, NW_006491746.1, NW_006490161.1, NW_006433953.1, NW_006476394.1, NC_023122.1,

NW_006441948.1, NW_006430464.1, NW_006491005.1, NW_006442104.1, NW_006433374.1, NW_006435837.1, NW_006492031.1, NW_006435017.1, NW_006438126.1, NW_006435844.1, NW_006436299.1, NW_006441386.1, NW_006489654.1, NW_006433716.1, NW_006435283.1, NW_006493951.1, NW_006438900.1, NW_006436669.1, NW_006436288.1, NW_006440646.1, NW_006438156.1, NW_006439717.1, NW_006494609.1, NW_006492006.1, NW_006494610.1, NW_006432451.1, NW_006442520.1, NW_006492654.1, NW_006492247.1, NW_006433527.1, NW_006442492.1, NW_006493151.1, NW_006442497.1, NW_006430785.1, NW_006494236.1, NW_006432452.1, NW_006441577.1, NW_006494316.1, NW_006442512.1, NW_006430554.1, NW_006432008.1, NW_006434516.1, NW_006430459.1, NW_006433947.1, NW_006442126.1, NW_006431394.1, NW_006491625.1, NW_006433948.1, NW_006438076.1, NW_006455683.1, NW_006436637.1, NW_006435759.1, NW_006443121.1, NW_006494561.1, NW_006433949.1, NW_006492718.1, NW_006430090.1, NW_006492919.1, NW_006494172.1, NW_006434832.1, NW_006431108.1, NW_006489397.1, NW_006429651.1, NW_006439715.1, NW_006437742.1, NW_006441837.1, NW_006439757.1, NW_006432265.1, NW_006440157.1, NW_006443512.1, NW_006430739.1, NW_006434904.1, NW_006430458.1, NW_006477065.1, NW_006436300.1, NW_006438868.1, NW_006494325.1, NW_006441076.1, NW_006436888.1, NW_006433124.1, NW_006494395.1, NW_006488845.1, NW_006481218.1, NW_006437784.1, NW_006488882.1, NW_006442109.1, NW_006494612.1, NW_006441394.1, NW_006438088.1, NW_006484309.1, NW_006431933.1, NW_006433019.1, NW_006443568.1, NW_006484869.1, NW_006434712.1, NW_006494594.1, NW_006431919.1, NW_006443567.1, NW_006440645.1, NW_006440214.1, NW_006441063.1, NW_006431074.1, NW_006436613.1, NW_006429611.1, NW_006432309.1, NW_006432596.1, NW_006431393.1, NW_006433386.1, NW_006443565.1, NW_006494483.1,

NW_006433879.1, NW_006442531.1, NW_006439199.1, NW_006490563.1, NW_006432699.1, NW_006429963.1, NW_006494597.1, NW_006443569.1, NW_006442103.1, NW_006493426.1, NW_006492864.1, NW_006440652.1, NW_006431079.1, NW_006432496.1, NW_006443120.1, NW_006430726.1, NW_006438644.1, NW_006494576.1, NW_006429330.1, NW_006434369.1, NW_006440647.1, NW_006494614.1, NW_006440832.1, NW_006433510.1, NW_006442124.1, NW_006490871.1, NW_006436815.1, NW_006431925.1, NW_006483424.1, NW_006431910.1, NW_006441068.1, NW_006442482.1, NW_006430465.1, NW_006441564.1, NW_006439919.1, NW_006436812.1, NW_006435779.1, NW_006430732.1, NW_006440152.1, NW_006436282.1, NW_006436746.1, NW_006434840.1, NW_006433950.1, NW_006443133.1, NW_006438077.1, NW_006434824.1, NW_006435909.1, NW_006490674.1, NW_006441441.1, NW_006436809.1, NW_006430496.1, NW_006441949.1, NW_006494611.1, NW_006491636.1, NW_006434827.1, NW_006488738.1, NW_006488725.1, NW_006443132.1, NW_006493320.1, NW_006443130.1, NW_006438086.1, NW_006435762.1, NW_006429388.1, NW_006436294.1, NW_006436301.1, NW_006430043.1, NW_006439713.1, NW_006492330.1, NW_006440626.1, NW_006434826.1, NW_006436808.1, NW_006488362.1, NW_006493721.1, NW_006440914.1, NW_006440618.1, NW_006439189.1, NW_006429381.1, NW_006490708.1, NW_006430240.1, NW_006487246.1, NW_006492320.1, NW_006494016.1, NW_006441067.1, NW_006443123.1, NW_006491957.1, NW_006494603.1, NW_006435760.1, NW_006443262.1, NW_006490699.1, NW_006431210.1, NW_006433385.1, NW_006429773.1, NW_006436813.1, NW_006494429.1, NW_006489710.1, NW_006492166.1, NW_006440160.1, NW_006440117.1, NW_006474108.1, NW_006431956.1, NW_006441435.1, NW_006443119.1, NW_006443471.1, NW_006433963.1, NW_006494412.1, NW_006435775.1, NW_006494447.1, NW_006433378.1, NW_006440461.1, NW_006431396.1,

NW_006440820.1, NW_006494592.1, NW_006433024.1, NW_006490257.1, NC_002619.1, NW_006440648.1, NW_006494445.1, NW_006441071.1, NW_006435754.1, NW_006442554.1, NW_006438873.1, NW_006436292.1, NW_006439219.1, NW_006493340.1, NW_006439194.1, NW_006484499.1, NW_006434367.1, NW_006437382.1, NW_006435869.1, NW_006480506.1, NW_006435849.1, NW_006440605.1, NW_006442106.1, NW_006436287.1, NW_006434835.1, NW_006438291.1, NC_026542.1, NW_006440131.1, NW_006442491.1, NW_006443292.1, NW_006439296.1, NW_006442564.1, NW_006429097.1, NW_006475342.1, NW_006437828.1, NW_006440625.1, NW_006492289.1, NW_006441525.1, NW_006431928.1, NW_006431046.1, NW_006431043.1, NW_006433946.1, NW_006431107.1, NW_006442921.1, NW_006494572.1, NW_006439190.1, NW_006493387.1, NW_006436373.1, NW_006431918.1

Bat Virus Accession Numbers: EF157976.1, GU170201.1, DQ837641.1, DQ648858.1, KP100644.1, AF369024.2, KC676792.1, FJ905105.2, Y09762.1, AF081020.2, JF311903.1, EF614258.1, EF065505.1, GU190215.1, KF636752.1, AF086833.2, EU293108.1, EF203064.1, EU420137.1, AF326114.2, HQ660129.1, JN899075.1, AF189155.1, AF212302.2, AY274119.3, JQ001749.1, JF828358.1, HQ595342.1, EF065509.1, NC_001474.2, JQ989270.1, EU420138.1, EF065513.1, EF157977.2, KF430219.1, EU420139.1, AF285080.1, DQ837641.1, HQ595340.1, M13215.1, HQ595344.1, DQ648794.1 , CY125942, CY103890, CY103873

Duck Accession Numbers: NW_004677685.1, NW_004676600.1, NW_004677281.1, NW_004677199.1, NW_004677172.1, NW_004676466.1, NW_004678549.1, NW_004684203.1, NW_004680828.1, NW_004677515.1, NW_004678089.1, NW_004739530.1, NW_004678473.1, NW_004677992.1, NW_004725899.1, NW_004676626.1, NW_004677274.1, NW_004676850.1, NW_004676363.1, NW_004676791.1, NW_004676880.1, NW_004677096.1, NW_004678065.1,

NW_004677317.1, NW_004676665.1, NW_004678935.1, NW_004677240.1, NC_009684.1, NW_004676692.1, NW_004683518.1, NW_004677664.1, NW_004690468.1, NW_004678474.1, NW_004677018.1, NW_004679201.1, NW_004749173.1, NW_004683443.1, NW_004676461.1, NW_004678414.1, NW_004678670.1, NW_004676471.1, NW_004676730.1, NW_004676473.1, NW_004679962.1, NW_004678345.1, NW_004678346.1, NW_004739768.1, NW_004677887.1, NW_004678274.1, NW_004753456.1, NW_004683406.1, NW_004677116.1, NW_004677232.1, NW_004676459.1, NW_004676775.1, NW_004677537.1, NW_004676891.1, NW_004743681.1, NW_004678215.1, NW_004677942.1, NW_004676893.1, NW_004754103.1, NW_004676716.1, NW_004690237.1, NW_004676785.1, NW_004677793.1, NW_004679555.1, NW_004676799.1, NW_004676963.1, NW_004677565.1, NW_004676436.1, NW_004679106.1, NW_004677015.1, NW_004676394.1, NW_004676773.1, NW_004676697.1, NW_004677126.1, NW_004676484.1, NW_004683332.1, NW_004676582.1, NW_004677476.1, NW_004678608.1, NW_004676580.1, NW_004680899.1, NW_004676800.1, NW_004745641.1, NW_004677132.1, NW_004676817.1, NW_004678275.1, NW_004676420.1, NW_004684118.1, NW_004689648.1, NW_004683645.1, NW_004679480.1, NW_004678221.1, NW_004683660.1, NW_004676336.1, NW_004677946.1, NW_004754515.1, NW_004743731.1, NW_004677373.1, NW_004677287.1, NW_004676696.1, NW_004682268.1, NW_004677175.1, NW_004676427.1, NW_004676738.1, NW_004679800.1, NW_004676369.1, NW_004678523.1, NW_004683061.1, NW_004676544.1, NW_004677655.1, NW_004678225.1, NW_004677068.1, NW_004677318.1, NW_004677184.1, NW_004752942.1, NW_004676592.1, NW_004676760.1, NW_004677254.1, NW_004677004.1, NW_004677527.1, NW_004677462.1, NW_004676627.1, NW_004677689.1, NW_004677973.1, NW_004676924.1, NW_004678318.1, NW_004678181.1, NW_004745447.1, NW_004677906.1, NW_004690505.1,

NW_004679235.1, NW_004676589.1, NW_004676715.1, NW_004689896.1, NW_004678002.1, NW_004676500.1, NW_004678462.1, NW_004677049.1, NW_004678279.1, NW_004676955.1, NW_004678259.1, NW_004678544.1, NW_004677470.1, NW_004677747.1, NW_004676552.1, NW_004676721.1, NW_004678669.1, NW_004677948.1, NW_004677774.1, NW_004676400.1, NW_004677091.1

Duck Virus Accession Numbers: EU910942.1, KJ000696.1, KC663628.1, JX987283.1, DQ226541.1, AY029299.1, CY181373, CY091590, CY012826, CY101938, CY095228, CY187158, KM244079, CY004539, EU026116, CY137481, CY032205, EU742640, EU743306, CY012804, EU743167, CY180007, KF424099, EU735790, CY145929, CY180793, KJ764739

Chapter 3

Conclusion

In this thesis, I have explored machine learning methods for the analysis of infectious disease dynamics and viral host identification. It was found that though regression is difficult, the time savings of a successful ML solution over the current state-of-the-art solution make it well worth improvement and application. Additionally, I found that classification is easier to implement in general, but further research is required to make it useful to the scientific community in general.

Publications

Tessmer, H. L., Ito, K., & Omori, R. (2018). Can machines learn respiratory virus epidemiology?: A comparative study of likelihood-free methods for the estimation of epidemiological dynamics. Frontiers in Microbiology, 9, 343.

Sakon, N., Komano, J., **Tessmer, H. L.**, & Omori, R. (2018). High transmissibility of norovirus among infants and school children during the 2016/17 season in Osaka, Japan. Eurosurveillance, 23(6), 18-00029.

Wang, J., Yu, X., **Tessmer, H. L.**, Kuniya, T., & Omori, R. (2017). Modelling infectious diseases with relapse: a case study of HSV-2. Theoretical Biology and Medical Modelling, 14(1), 13.

Nyirenda, M., Omori, R., **Tessmer, H. L.**, Arimura, H., & Ito, K. (2016). Estimating the lineage dynamics of human influenza B viruses. PloS one, 11(11), e0166107.

Karnbunchob, N., Omori, R., **Tessmer, H. L.**, & Ito, K. (2016). Tracking the evolution of polymerase genes of influenza a viruses during interspecies transmission between avian and swine hosts. Frontiers in microbiology, 7, 2118.

Publications

Omori, R., Nakata, Y., **Tessmer, H. L.**, Suzuki, S., & Shibayama, K. (2015). The determinant of periodicity in Mycoplasma pneumoniae incidence: an insight from mathematical modelling. Scientific reports, 5, 14473.

Presentations

Tessmer, HL, Omori, R. 'Estimation of Basic Reproduction Number R0 using a Recurrent Neural Network' in the Workshop on Machine Learning for Health, NIPS (Neural Information Processing Systems). December 5-10, 2016. Barcelona, Spain.

Tessmer, HL, Ito, K. 'Computational Analyses of the Influenza and Ebola Viral Genomes'. SaSSOH (Sapporo Summer Seminar for One Health). September 16-17, 2015. Sapporo, Japan.

Tessmer, HL, Ito, K. 'Computational Analyses of the Influenza and Ebola Viruses'. SMBE (Society for Molecular Biology and Evolution). July 12-16, 2015. Vienna, Austria.

Tessmer, HL, Ito, K. Tracking Influenza Epidemics using Bioinformatics Techniques. Genome Informatics Workshop (GIW ISCB-ASIA). December 15-17, 2014. Tokyo, Japan.

Tessmer, HL, Ito, K. 'Computational Analyses of the Influenza and Ebola Viral Genomes'. SaSSOH (Sapporo Summer Seminar for One Health). September 24-25, 2014. Sapporo, Japan.

References

[1] Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (2010). Theano: A cpu and gpu math compiler in python. In *Proc. 9th Python in Science Conf*, pages 1–7.

[2] Berthold, M. and Hand, D. J. (2003). *Intelligent data analysis: an introduction*. Springer Science & Business Media.

[3] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

[4] Bjørnstad, O. N., Finkenstädt, B. F., and Grenfell, B. T. (2002). Dynamics of measles epidemics: Estimating scaling of transmission rates using a time series sir model. *Ecological Monographs*, 72(2):169–184.

[5] Caicedo, J. P. R., Verrelst, J., Muñoz-Marí, J., Moreno, J., and Camps-Valls, G. (2014). Toward a semiautomatic machine learning retrieval of biophysical parameters. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(4):1249–1259.

References

[6] Calisher, C. H., Childs, J. E., Field, H. E., Holmes, K. V., and Schountz, T. (2006). Bats: important reservoir hosts of emerging viruses. *Clinical microbiology reviews*, 19(3):531–545.

[7] Chaplot, S., Patnaik, L., and Jagannathan, N. (2006). Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network. *Biomedical Signal Processing and Control*, 1(1):86 – 92.

[8] Cheng, X., Virk, N., Chen, W., Ji, S., Ji, S., Sun, Y., and Wu, X. (2013). Cpg usage in rna viruses: data and hypotheses. *PloS one*, 8(9):e74109.

[9] Cooke, K. L. and van den Driessche, P. (1996). Analysis of an seirs epidemic model with two delays. *Journal of Mathematical Biology*, 35(2):240–260.

[10] Diekmann, O., Heesterbeek, J., and Roberts, M. (2009). The construction of next-generation matrices for compartmental epidemic models. *Journal of the Royal Society Interface*, page rsif20090386.

[11] Diekmann, O., Heesterbeek, J. A. P., and Metz, J. A. (1990). On the definition and the computation of the basic reproduction ratio r0 in models for infectious diseases in heterogeneous populations. *Journal of mathematical biology*, 28(4):365–382.

[12] Dietterich, T. G. et al. (2000). Ensemble methods in machine learning. *Multiple classifier systems*, 1857:1–15.

[13] Dreiseitl, S. and Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: a methodology review. *Journal of Biomedical Informatics*, 35(5):352 – 359.

[14] Duffy, S., Shackelton, L. A., and Holmes, E. C. (2008). Rates of evolutionary change in viruses: patterns and determinants. *Nature Reviews Genetics*, 9(4):267.

[15] Edmunds, W. J., Gay, N. J., Kretzschmar, M., Pebody, R. G., and Wachmann, H. (2000). The pre-vaccination epidemiology of measles, mumps and rubella in europe: implications for modelling studies. *Epidemiology and Infection*, 125(3):635–650.

[16] Fraser, C., Donnelly, C. A., Cauchemez, S., Hanage, W. P., Van Kerkhove, M. D., Hollingsworth, T. D., Griffin, J., Baggaley, R. F., Jenkins, H. E., Lyons, E. J., Jombart, T., Hinsley, W. R., Grassly, N. C., Balloux, F., Ghani, A. C., Ferguson, N. M., Rambaut, A., Pybus, O. G., Lopez-Gatell, H., Alpuche-Aranda, C. M., Chapela, I. B., Zavala, E. P., Guevara, D. M. E., Checchi, F., Garcia, E., Hugonnet, S., Roth, C., and (2009). Pandemic potential of a strain of influenza a (h1n1): Early findings. *Science*, 324(5934):1557–1561.

[17] Greenbaum, B. D., Levine, A. J., Bhanot, G., and Rabadan, R. (2008). Patterns of evolution and host gene mimicry in influenza and other rna viruses. *PLoS pathogens*, 4(6):e1000079.

[18] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

[19] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France. PMLR.

[20] Jenkins, G. M., Pagel, M., Gould, E. A., Paolo, M. d. A., and Holmes, E. C. (2001). Evolution of base composition and codon usage bias in the genus flavivirus. *Journal of molecular evolution*, 52(4):383–390.

[21] Jourdain, E., Gunnarsson, G., Wahlgren, J., Latorre-Margalef, N., Bröjer, C., Sahlin, S., Svensson, L., Waldenström, J., Lundkvist, Å., and Olsen, B. (2010). Influenza virus in a natural host, the mallard: experimental infection data. *PloS one*, 5(1):e8935.

[22] Karlin, S., Doerfler, W., and Cardon, L. (1994). Why is cpg suppressed in the genomes of virtually all small eukaryotic viruses but not in those of large eukaryotic viruses? *Journal of virology*, 68(5):2889–2897.

[23] Karlin, S. and Mrázek, J. (1996). What drives codon choices in human genes? *Journal of molecular biology*, 262(4):459–472.

[24] Kolen, J. F. and Pollack, J. B. (1991). Back propagation is sensitive to initial conditions. In *Advances in neural information processing systems*, pages 860–867.

[25] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and

Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

[26] Leendertz, S. A. J., Gogarten, J. F., Düx, A., Calvignac-Spencer, S., and Leendertz, F. H. (2016). Assessing the evidence supporting fruit bats as the primary reservoirs for ebola viruses. *EcoHealth*, 13(1):18–25.

[27] Lessler, J., Reich, N. G., Cummings, D. A., the New York City Department of Health, and Team, M. H. S. I. I. (2009). Outbreak of 2009 pandemic influenza a (h1n1) at a new york city school. *New England Journal of Medicine*, 361(27):2628–2636. PMID: 20042754.

[28] Lobo, F. P., Mota, B. E., Pena, S. D., Azevedo, V., Macedo, A. M., Tauch, A., Machado, C. R., and Franco, G. R. (2009). Virus-host coevolution: common patterns of nucleotide motif usage in flaviviridae and their hosts. *PloS one*, 4(7):e6282.

[29] Ma, H., El-Keib, A. A., and Ma, X. (1994). Training data sensitivity problem of artificial neural network-based power system load forecasting. In *Proceedings of 26th Southeastern Symposium on System Theory*, pages 650–652.

[30] Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30.

[31] Magal, P. and Ruan, S. (2014). Susceptible-infectious-recovered models revisited: From the individual level to the population level. *Mathematical Biosciences*, 250:26 – 40.

[32] Mossong, J. and Muller, C. (2000). Estimation of the basic reproduction number of measles during an outbreak in a partially vaccinated population. *Epidemiology & Infection*, 124(2):273–278.

[33] Nishiura, H., Castillo-Chavez, C., Safan, M., Chowell, G., et al. (2009). Transmission potential of the new influenza a (h1n1) virus and its age-specificity in japan. *Euro Surveill*, 14(22):19227.

[34] Palmer, J. and Chakravarty, A. (2014). Supervised machine learning. *An Introduction To High Content Screening: Imaging Technology, Assay Development, and Data Analysis in Biology and Drug Discovery*, page 231.

[35] Sahiner, B., Chan, H.-P., Petrick, N., Wei, D., Helvie, M. A., Adler, D. D., and Goodsitt, M. M. (1996). Classification of mass and normal breast tissue: a convolution neural network classifier with spatial domain and texture images. *IEEE Transactions on Medical Imaging*, 15(5):598–610.

[36] Saulnier, E., Gascuel, O., and Alizon, S. (2017). Inferring epidemiological parameters from phylogenies using regression-abc: A comparative study. *PLOS Computational Biology*, 13(3):1–31.

[37] Shackelton, L. A. and Holmes, E. C. (2004). The evolution of large dna viruses: combining genomic information of viruses and their hosts. *Trends in microbiology*, 12(10):458–465.

[38] Sokolova, M., Japkowicz, N., and Szpakowicz, S. (2006). *Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation*, pages 1015–1021. Springer Berlin Heidelberg, Berlin, Heidelberg.

[39] Sullivan, K. M., Halpin, T. J., Marks, J. S., and Kim-Farley, R. (1985). Effectiveness of mumps vaccine in a school outbreak. *American Journal of Diseases of Children*, 139(9):909–912.

[40] Sunnåker, M., Busetto, A. G., Numminen, E., Corander, J., Foll, M., and Dessimoz, C. (2013). Approximate bayesian computation. *PLOS Computational Biology*, 9(1):1–10.

[41] Sved, J. and Bird, A. (1990). The expected equilibrium of the cpg dinucleotide in vertebrate genomes under a mutation model. *Proceedings of the National Academy of Sciences*, 87(12):4692–4696.

[42] Tong, S., Li, Y., Rivailler, P., Conrardy, C., Castillo, D. A. A., Chen, L.-M., Recuenco, S., Ellison, J. A., Davis, C. T., York, I. A., et al. (2012). A distinct lineage of influenza a virus from bats. *Proceedings of the National Academy of Sciences*, 109(11):4269–4274.

[43] van Hemert, F., van der Kuyl, A. C., and Berkhout, B. (2014). On the nucleotide composition and structure of retroviral rna genomes. *Virus research*, 193:16–23.

[44] Verrelst, J., Muñoz, J., Alonso, L., Delegido, J., Rivera, J. P., Camps-Valls, G., and Moreno, J. (2012). Machine learning regression algorithms for biophysical parameter retrieval: Opportunities for sentinel-2 and -3. *Remote Sensing of Environment*, 118:127 – 139.

[45] Vorou, R., Papavassiliou, V., and Tsiodras, S. (2007). Emerging zoonoses and vector-borne infections affecting humans in europe. *Epidemiology & Infection*, 135(8):1231–1247.

[46] Vynnycky, E., Trindall, A., and Mangtani, P. (2007). Estimates of the reproduction numbers of spanish influenza using morbidity data. *International Journal of Epidemiology*, 36(4):881–889.