Species Delimitation by Neural Networks Using Multi-locus Sequence Data

by

Qifang Pang

(Under the Direction of Liang Liu)

Abstract

In biology, species is the fundamental unit of classification of organisms that share common characteristics and are able to interbreed. Determining species boundaries (i.e., species delimitation) is crucial for studies in ecology, evolutionary biology, and other biological fields. The biological definition of species based on reproductive isolation, however, is difficult to test using empirical data. Recent development of phylogenetic species provides a conceptual framework of stochastic models for species delimitation, in which species are defined as lineages in a phylogenetic tree. Researchers focusing on complex speciation scenarios during evolutionary process have shown that phylogenetic tools can accurately identify the boundaries of cryptic species. However, these methods are often time-consuming and fail to handle a massive amount of molecular sequence data. High performance of deep learning algorithms has inspired scientists to develop computationally efficient tools for species delimitation using DNA sequences. In this project, we apply neural networks algorithms for species delimitation using multi-locus sequence data. The performance of the proposed method is compared with other machine learning and model-based approaches. The results of simulation and real data analyses show that neural network outperforms the other approaches in delimitating species boundaries.

Index words: Neural Networks, Species Delimitation, Deep Learning, Phylogenetics

Species Delimitation by Neural Networks Using Multi-locus Sequence Data

by

Qifang Pang

B.S., Huazhong University of Science and Technology, 2019

A Thesis Submitted to the Graduate Faculty of the
University of Georgia in Partial Fulfillment of the Requirements for the Degree.

Master of Science

Athens, Georgia

2021

Species Delimitation by Neural Networks Using Multi-locus Sequence Data

by

Qifang Pang

Major Professor:	Liang Liu
Committee:	Yuan Ke
	Jaxk H Reeves

Electronic Version Approved:

Ron Walcott
Vice Provost for Graduate Education and Dean of the Graduate School
The University of Georgia
August 2021

Acknowledgments

Firstly, I would like to thank my advisor Dr. Liang Liu from the department of statistics at the University of Georgia. His guidance helped me finish my project. I could not have finished the thesis without the support of Dr. Liu. Secondly, I would like to thank to my committee members: Dr. Jaxk H Reeves and Dr. Yuan Ke , for their insightful comments and advice. Finally, I would like to thank my parents and friends for their help and support throughout this project.

Contents

Acknowledgments **iv**

List of Figures **vi**

List of Tables **viii**

1 Introduction **1**

1.1 Introduction 1

1.2 Literature Review 2

2 Methods **7**

2.1 Method 7

2.2 GMYC 12

2.3 Machine Learning Approaches 13

3 Empirical data sets and Simulations **14**

3.1 Sequence Generate Procedure 14

3.2 Sim A 16

3.3 Sim B 17

3.4 Sim C 19

3.5 Empirical data sets 20

4 Analysis and Results 24
4.1 Mathematical Statistics Tools 24
4.2 Sim A 25
4.3 Sim B 28
4.4 Sim C 30
4.5 Empirical Data Sets 32

5 Discussion 33

Appendices 35

A 35

Bibliography 39

List of Figures

2.1 Sequence be coded as a matrix. 8

2.2 Example of one-hot encode on a DNA sequence, let $A = [1, 0, 0, 0], C = [0, 1, 0, 0], G = [0, 0, 1, 0], T = [0, 0, 0, 1]$, a sequence of length n is then represent by a numerical vector of length 4n 9

2.3 Visualization of our neural network model. The size of input layer is 4n in which n is the length of our input DNA sequence, 4 hidden layers are added in our model. The output layer has four neurons representing four different species. 10

3.1 Flow chart of sequence generating process. 15

3.2 Visualization of species tree in simulation A. 16

3.3 Visualization of species tree in simulation B. 18

3.4 Visualization of species tree in simulation C and D. 19

3.5 Phylogeny of sand dragon population. 21

3.6 Phylogeny of Geeko population. 23

4.1 Barplot of accuracy table 4.1 26

4.2 Confidence interval for neural network in SimA 27

4.3 Barplot of accuracy table 4.2 28

4.4 Confidence interval for neural network in SimB 29

4.5 Barplot of accuracy table 4.3 31

A.1 Visualization of accuracy table 4.1 35

A.2 Visualization of accuracy table 4.2 36

A.3 Visualization of accuracy table 4.3 37

A.4 Standard deviation plot for neural network classifier 38

List of Tables

4.1 Accuracies for synthetic datasets of simulation A[%]. 26

4.2 Accuracies for synthetic datasets of simulation B [%]. 29

4.3 Accuracies for synthetic datasets of simulation C [%]. 30

4.4 Accuracies on empirical data sets [%]. 32

Chapter 1

Introduction

1.1 Introduction

In biology, Species are fundamental units in most biological subdisciplines. In the field of evolutionary genetics, species are the most widely used terminal taxa in reconstruction of phylogenetic trees. In systematics, species are the basic taxonomic units to study the biological diversity and relationships among organisms. In conservation biology, since biodiversity hot spots are established in terms of the number of species, inaccurate species classification may severely change the biodiversity assessment, and subsequently affect conservation practice. Thus, accurate classification of organisms at the species level is crucial in all branches of biological fields.

Accurate identification of cryptic species is quite challenging (Z. Yang & Rannala, 2014), because morphological features are uninformative and provide little evidence to delimit species boundaries. Morphological data were first used to estimate species boundaries, but they suffer from numerous problems of missing characters, incompatibility among species, and lack of measurable features. In recent years, molecular sequences have become the primary data for delimiting species boundaries. Genetic data such as DNA and protein sequences are more informative than morphological data for understanding the evolutionary history of species. The accumulation of mutations during the evolutionary process of species triggered speciation events resulting in a split (i.e., two independent lineages) in the phylogenetic tree of species. Thus, between-species genetic variation should be significantly higher than with-in species variation, which can be used to identify species boundaries from molecular sequences. Due to the rapid advance of DNA sequencing techniques, it becomes much cheaper and easier to get whole genome sequence data for the organisms of interest than ever before. Meanwhile, statistical approaches have been developed to infer species boundaries from molecular sequence data.

1.2 Literature Review

1.2.1 Automatic Barcode Gap Discovery

In 2003, Herbert introduced DNA barcoding for species identification using the short sequences of mitochondrial genomes. (Hebert et al., 2003). Afterward, the sequences of other regions in the mitochondrial genome were utilized as effective molecular markers to identify species. These regions were chosen because they are less variant within species than between species, which is known as a barcoding gap (Meyer & Paulay, 2005). The Automatic Barcode Gap Discovery (ABGD) attempts to identify the barcode gap through an automatic process (Puillandre et al., 2012). This program first computes all pairwise sequence distances: for a group of n DNA sequences, $n(n-1)/2$ pairwise distances are computed and then the distances are ranked in ascending order. At rank r, a local slope function is calculated for the distance value d_r:

$$s_{r,w} = \frac{d_{r+w} - d_w}{w}$$

The program detects the largest value of the local slope and reports the distance at which the function takes its local maximum. The distance is the exact barcode gap that will be used to delimit species. It is notable that the barcode gap should be larger than a threshold value ($dist_{limit}$) under which the sequences are statistically significantly intraspecific. Simulations based on standard coalescent model are used to compute $dist_{limit}$. It was shown that if the population size n > 10, the threshold value $dist_{limit}$ has a linear relationship with population mutation rate θ, where θ can be estimated by calculating the average pairwise distance that is smaller than a user-defined prior intra-specific diversity. Once the value of θ is given $dist_{limit}$ is determined, a barcode gap can be also computed.

The program splits sequences dataset into small groups based on the barcode gap: for two sequences that are chosen from different groups, their distance must be larger than the gap distance. This partition procedure is then applied recursively to groups that are newly formed until no additional group is made.

While ABGD is a computationally efficient method and can be used to analyze large datasets, it relies on the premise that a barcode gap exists (Wiemers & Fiedler, 2007). However, studies have shown that such a limit does not exist for specific species, making the classification results questionable. Second, this method requires an appropriate prior threshold for intra-specific divergence. Different priors result in different species partitions. Third, ABGD is a distance-based method, thus information is lost when difference among nucleotides sequences are converted into genetic distances (Zhang et al., 2008). Finally, it should be kept in mind that species delimitation based on a single locus is not robust enough. The result can only be treated as a preliminary hypothesis on which further analysis should be carried out (Lukhtanov, 2019).

1.2.2 General Mixed Yule Coalescent

The General Mixed Yule Coalescent (GMYC) is a maximum likelihood method that requires an ultrametric gene tree as input. (Pons et al., 2006). The GMYC method determines diversification and coalescence events based on branching patterns. It attempts to detect a

Time (T) in the tree at which the process of node switches from a Yule speciation to the intra-species coalescent process. Species delimitation is determined by T which is computed using the maximum likelihood method.

Studies have found that the accuracy of the method is affected by factors such as the product of speciation rate and effective population size, the ratio of average population size to divergence time (Esselstyn et al., 2012; Fujisawa & Barraclough, 2013). The method also requires its input tree to be strictly ultra-metric and bifurcating which is a difficult task. We need to apply a genealogy-based inference approach to generate an ultrametric tree which greatly increasing computational time. Moreover, ultra-metric gene tree constructed from single-locus data is unreliable to make predictions. as we have said earlier, the delimitation output based on single-locus data by GMYC method is not solid. It should be interpreted as a species classification hypothesis on which further work should be carried out(Luo et al., 2018).

Implementation of this method is extremely time-consuming: to obtain an ultrametric tree, we first use BEAUti software to generate a xml file. BEAUti does not allow concurrent input and out. One must input the file from the folder, set desirable parameters, and then name the output file. The xml file is then used as input of Beast (Suchard et al., 2018) program to generate a tree file. This process usually required ten to forty minutes depends on the length and number of our sequence. Finally, R package Splits (Ezard et al., 2009) is utilized to do species delimitation.

1.2.3 Bayesian Phylogenetics and Phylogeography

Bayesian Phylogenetics and Phylogeography (BPP) is a Bayesian Markov Chain Monte Carlo (MCMC) program for analyzing DNA sequence alignments under the multispecies coalescent model (MSC) (Rannala & Yang, 2003; Z. Yang & Rannala, 2010). In contrast to GMYC and PTP methods, BPP program is designed to delimitate species boundaries using multi-locus DNA sequence data. This method is more reliable to some extent, but it is also more computationally expensive (Z. Yang, 2015). Thus, a user-specific phylogenetic tree (guide tree) that represents the genetic relation of DNA sequences is used to reduce computational time (Z. Yang & Rannala, 2010). The algorithm will identify all species delimitation models that are consistent with the guided tree, then posterior probabilities for each model are computed. Although this Bayesian method has a very low error rate in most of the simulation studies, research conducted by Zhang, et al. (2011) suggests that this method will perform poorly if gene flow between species present.

1.2.4 Machine Learning Approaches

Significant progress has been made in the direction of using machine learning and deep learning techniques to solve classification problems. It has been proven that the machine learning algorithms below are effective to classify DNA barcodes of various organisms(Weitschek et al., 2014). The classification results show that machine learning methods are promising tools for handling nucleotides sequences. But how well do these methods classify multilocus data? Are there any other approaches that perform better than these methods? These questions inspired us to apply deep learning algorithms to classify genetic data.

Support Vector Machines (SVM) is a discriminative classifier (Platt, 1999), It converts the reference DNA sequences into multi-dimensional vectors and defines a separating hyperplane among the sequences belonging to different classes. For example, given labeled training data, the algorithm outputs an optimal hyperplane that separates the classes with the largest minimum distance. After a proper vector transformation, new objects from the query set are evaluated according to this separating hyperplane. One of the most relevant features of the SVM is to use a non-linear transformation of the input data in a very efficient way by using a linear kernel function. SVM performs usually with high classification accuracy, but the main drawback is that the output classification model is hard to interpret.

A decision tree (Salzberg, 1994) is a simple tree structure in which non-terminal vertices represent tests on attributes, and the terminal vertices reflect the results of the decision. Decision trees are simple and easily converted into a set of rules, it can handle both numerical and categorical data (even if the output attribute must be categorical). However, decision trees are unstable, variations in the training data can produce a different set of attributes, and it is not allowed to produce multiple output attributes. The classification model (the decision tree) can be easily read as a set of logic rules formed by sequence positions and nucleotide assignments. In our study, we apply the supervised machine learning method J48. It is implemented by the decision tree algorithm C4.5 on Weka platform (Quinlan, 1993; Witten & Frank, 2005).

Naive Bayes is a Bayesian-based classifier using estimator classes (Wang et al., 2007). It is a practical learning method often used when a large reference set is available. The critical assumption of a Naive Bayes classifier is that an attribute is independent of any other features. Based on this assumption, a Bayesian classifier is computed: based on the observable variables and prior probabilities, the posterior probabilities of the unknown attributes are evaluated. In this way, we can use the classifier as a tool of investigation and forecasting. As with the SVM method, the classification model provided by naive Bayes method is hard to interpret, we can assign the specimen to species only ”blindly”.

Chapter 2

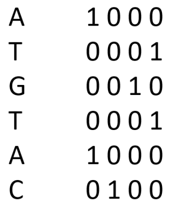
Methods

2.1 Method

2.1.1 Choice of Our Method

In the beginning, Convolutional Neural Networks (CNN) was considered as our candidate model since this method has been proved effective in image recognition. Generally, A three-dimensional array is commonly used to represent images in computers; the first dimension is the number of layers (Red, Green, Blue), the second and third dimensions are the image's height and width. A multi-locus sequence can also be represented as a three-dimensional array. The sequence from a single locus can be code as in figure 2.1. We can also get a three-dimensional array by stacking matrices coded from different loci. We used TensorFlow framework to build (Abadi et al., 2015) our model. Contrary to our expectations, the model did not perform well; several different parameter settings were used, but CNN still achieved low accuracy.

Neural networks (NNs) have been used to classify sequences (Zhang et al., 2008). Generally, a neural network required its input data as a real-valued vector. Thus, the previous study converts A, T, G, C to 0.1, 0.2, 0.3, 0.4 respectively to transform a DNA sequence to a



A	1 0 0 0
T	0 0 0 1
G	0 0 1 0
T	0 0 0 1
A	1 0 0 0
C	0 1 0 0

Figure 2.1: Sequence be coded as a matrix.

real-valued vector (A, T, G, C represent nucleotide bases Adenine (A), Cytosine (C), Guanine (G), Thymine (T)). However, such a transformation is inappropriate. A categorical variable can be either nominal or ordinal. The variable composed of four nucleotide bases should be treated as a nominal variable. However, if A, T, G, C were transformed to real-value 0.1, 0.2, 0.3, 0.4 respectively, they are ordered.

The neural network model is calculated based on metrics in the vector space, so we want to transform A, T, G, C, to non-partially ordered variables have no partial order. Using one-hot coding, the value of the discrete feature is extended to the Euclidean space, and a certain value of the discrete feature corresponds to a point in the Euclidean space. Using one-hot encoding for discrete features will make the distance calculation between features more reasonable. Hence, we apply one-hot encode approach in our project. The next section introduces details about one-hot encode.

The classifier used in the previous study has another drawback. This classifier utilizes a sigmoid activation function which tends to vanish gradients, then the model was unable to ”learn” from data. In our study, a different neural network was implemented to solve the above questions. Our model obtained a higher accuracy compared to other machine learning methods.

2.1.2 One-hot Encoding

Neural networks required their training data to consist of a numerical vector x and a numerical label y. Thus, DNA data composed of four single letters: A (adenine), G (guanine), C (cytosine), and T (thymine) need to be transformed to numbers. In the beginning, a K-mer encoding is considered. A k-mer is a word of length k using the four letters of the DNA alphabet. K-mer encoding works by establishing a k-mer "library" contains all possible words of length k and counting how many times each word occurs in each sequence. There are 4^k different kinds of k-mer. The K-mer encoding can thus transform a data set in which sequences are of different lengths into a data set where vectors all have an equal length 4^k. It is desirable for an artificial neural network that required its input of the same length. However, the transformation for a long sequence is time-consuming. Moreover, if k=9 was selected, a sequence is then transformed into a vector of length $4^9 = 262144$, such an input significantly increased neural network computational time.

Hence, we apply one-hot encoding where A, C, G, T are mapped onto $[1, 0, 0, 0]$, $[0, 1, 0, 0]$, $[0, 0, 1, 0]$, $[0, 0, 0, 1]$, respectively. Figure 2.2 illustrates how a DNA sequence is represented by a numerical vector. For our real data sets. Sequences also contained uncertain nucleotides denote by "?" or "-", we code them as [0.5, 0.5, 0.5, 0.5].

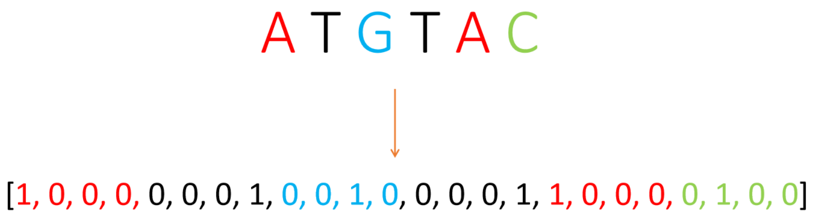


Figure 2.2: Example of one-hot encode on a DNA sequence, let $A = [1, 0, 0, 0], C = [0, 1, 0, 0], G = [0, 0, 1, 0], T = [0, 0, 0, 1]$, a sequence of length n is then represent by a numerical vector of length 4n

2.1.3 General layout

A neural network is a parallel computational model consisting of a large number of processing units (neurons) that are tied together with weighted connections.(Wu, 1997). The question is, how many processing units should be considered in our model? We test our model on validation data sets (validation data sets were simulated using the same procedure as described in chapter 3) to find the optimal parameters and a four-layer structure was chose to classify sequences. Figure 2.3 shows the neural network that was used in our study. The leftmost layer of this network is the input layer, and the neurons within the layer are called input neurons. The rightmost layer is the output layer containing the output neurons. Layers between the input layer and output layer are hidden layers. This neural network is a feed-forward neural network since its output from one layer is used as input of the next layer.

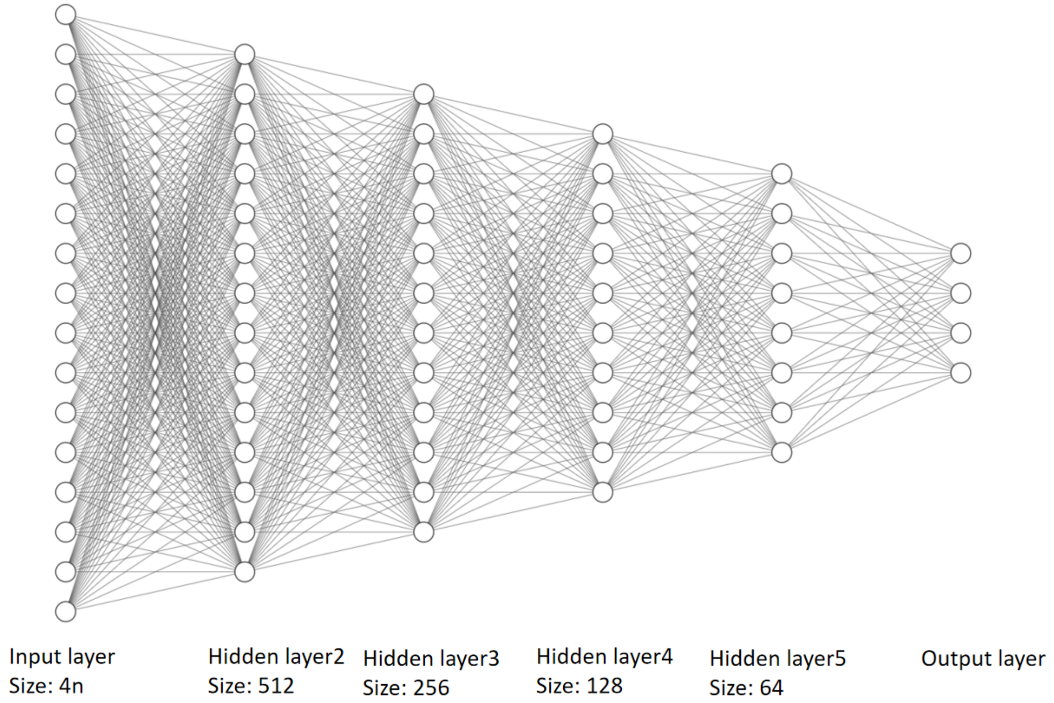


Figure 2.3: Visualization of our neural network model. The size of input layer is 4n in which n is the length of our input DNA sequence, 4 hidden layers are added in our model. The output layer has four neurons representing four different species.

The line connecting neurons between layers are weights, they can be described in matrix form:

$$W_{(l)} = \begin{pmatrix} w^l_{11} & \dots & w^l_{1n} \\ \vdots & \ddots & \vdots \\ w^l_{h1} & \cdots & w^l_{hn} \end{pmatrix}$$

where (w^l_{jk}) is weight for the connection from the k-th neuron in the $(l-1)$-th layer to the j-th neuron in the l-th layer. h and n is the number of neurons in l-th layer and $(l-1)$-th layer, respectively.

Hidden layers receive input from other layers and provide output for the next layer. The Relu function is used to compute the value of neurons in hidden layers, this function is called activation function and the value computed is activation. The Relu function is calculated as follows:

$$f(x) = max(0, x) \tag{1}$$

The activation a^l_j of the j-th neuron in the l-th layer is related to the activations in $(l-1)$-th layer by the equation 2:

$$a^l_j = f\left(\sum_{i=1,k} w^l_{ji} a^{l-1}_i\right) \tag{2}$$

k is number of neurons in $(l-1)$-th layer. $\sum_{i=1,k} w^l_{ji} a^{l-1}_i$ is the weight input of the j-th neuron in the l-th layer, denote as z^l_j.

Given the inputs and weights, we can compute activations of the first hidden layer, those values were then be used as input to calculate activations of next layer. Repeating this procedure the output value of last hidden layer will be found eventually. Then we use a softmax function to generate final output:

$$softmax(a)_i = \frac{\exp{(a_i)}}{\sum_{j=1}^{o} \exp{(a_j)}} \tag{3}$$

a_i represents the i-th value of the input vector in output layer, and o is the number of species. The result is a numerical vector in which elements correspond to probabilities that the sequence belongs to each species. The species with the highest probability is the output.

2.1.4 Implementation

One-hot encode of DNA sequence data and implementation of the seven layers neural network is done using the Python programming language (Van Rossum & Drake, 2009), utilizing the scikit-learn (Pedregosa et al., 2011) and Tensorflow (Abadi et al., 2015) frameworks for deep learning.

2.2 GMYC

The General Mixed Yule Coalescent (GMYC) is a maximum likelihood method designed to delineate species using sequence data from a single locus (Pons et al., 2006). Thus we use concatenated sequences to perform our analysis. This method requires an ultrametric gene tree in which all tips have the same age. We use software BEAST v1.10.4 (Suchard et al., 2018) to obtain this ultrametric tree. Since BEAST uses a Bayesian framework to do estimation, we need to set priors to infer the branch length and node times of the tree. We choose the Gamma site heterogeneity model and a constant coalescent. A relaxed clock that assumes that mutation rates of each branch vary over the tree and are drawn from a statistical distribution (Drummond et al., 2006). Here we will use a log-normal distribution which is similar to normal distribution but all the values are positive. The estimated sample size (ESS) values were checked once the tree was produced, the ESS values should be greater than 200 which indicates our parameters used were accurate. R package Splits (Ezard et al., 2009) is used to do species delimitation.

2.3 Machine Learning Approaches

We use Weka 3.8.5 (Witten & Frank, 2005) data mining platform to analyze our sequence data. The Weka platform requires input files to have a special format called ARFF, and DNA sequences are always stored in the standard FASTA format. We use the software Fasta2Weka (Weitschek et al., 2014) to convert FASTA format into ARFF format, this software does not support multifile conversion. So 750 datasets (750 datasets were simulated in chapter3) were transformed one by one.

Chapter 3

Empirical data sets and Simulations

3.1 Sequence Generate Procedure

We started from a typical species tree that contains four species: orangutan, chimpanzee, gorilla, and human, from which we derive multiple gene trees using the formula derived by (Rannala & Yang, 2003), this formula is applied by function sim.coaltree.sp in R package Phybase (Liu & Yu, 2010). Since each gene tree represents the evolutionary history of a single gene locus, we can use multiple gene trees to simulate multi-locus data. For each gene tree generated by Phybase, We used Seq-Gen v1.3.4 (Rambaut & Grassly, 1997) to simulate DNA sequences based on the Jukes–Cantor nucleotide substitution model (JUKES & CANTOR, 1969), with each nucleotide sequence 250 bases long.

we used each of three 4-taxon species tree to generate 1, 2, 4, 10, 20 gene trees under the multispecies coalescent model. Each gene tree contains 100 taxa (25 taxa per species, each taxon has a unique label). Then, from each gene tree, we used Seq-gen software to generate 100 DNA sequences of 250 base pairs. Concatenating DNA sequences simulated from different gene trees yields multi-locus DNA sequences. As a result, the N-locus dataset

(N is the gene number) contains 100 DNA sequences, each of which is 250N bases long. A training set of 80 sequences is randomly selected and used for fitting the 5 methods, the remaining 20 sequences are used to compute classification accuracy. Figure 3.1 illustrates how to simulate N-locus DNA sequences.

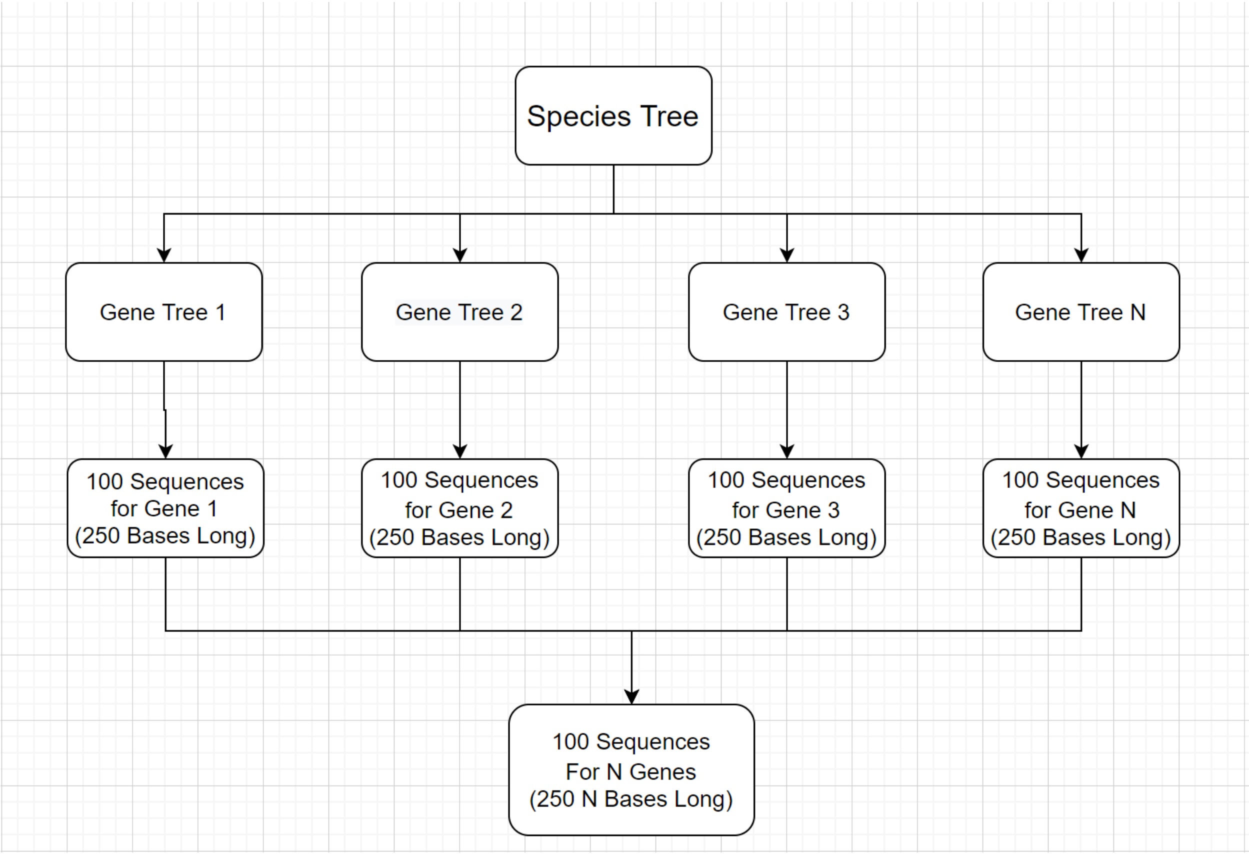

Species Tree
Gene Tree 1
Gene Tree 2
Gene Tree 3
Gene Tree N
100 Sequences for Gene 1 (250 Bases Long)
100 Sequences for Gene 2 (250 Bases Long)
100 Sequences for Gene 3 (250 Bases Long)
100 Sequences for Gene N (250 Bases Long)
100 Sequences For N Genes (250 N Bases Long)

Figure 3.1: Flow chart of sequence generating process.

We simulate 1-locus, 2-locus, 4-locus, 10-locus, and 20-locus DNA sequences for each of the three species trees. Thus, there are 15 different scenario combinations. Each scenario was played out 50 times. As a result, we have 750 data sets to test using five different methods. The average accuracy of 50 repeats is computed for each method.

Three species tree have the same tree topology and branch length: $(((H : 0.00402, C : 0.00402) : 0.00304, G : 0.00707) : 0.00929, O : 0.01635)$. But the parameter θ (effective population size multiplied by mutation rate) was set differently, resulting in nucleotide sequences with varying levels of classification difficulty. The following sections describe simulations for each of the three species trees.

3.2 Sim A

In simulation A we used the following tree: $(((H : 0.00402\ \#0.1, C : 0.00402\ \#0.1) : 0.00304\ \#0.001, G : 0.00707\ \#0.001) : 0.00929\ \#0.001, O : 0.01635\ \#0.001)\ \#0.001$. The values following the "#" sign are the mutation rate multiplied by the effective population size. This value is denoted by θ. Effective population size is the number of individuals in a population who give birth to the next generation. The greater the value θ, the less genetic divergence there is between two species. For example, the red branches is figure 3.2 have a large value 0.1, implying that the effective population size of the evolutionary process along two branches is large, resulting in less nucleotides sequence difference between two species.

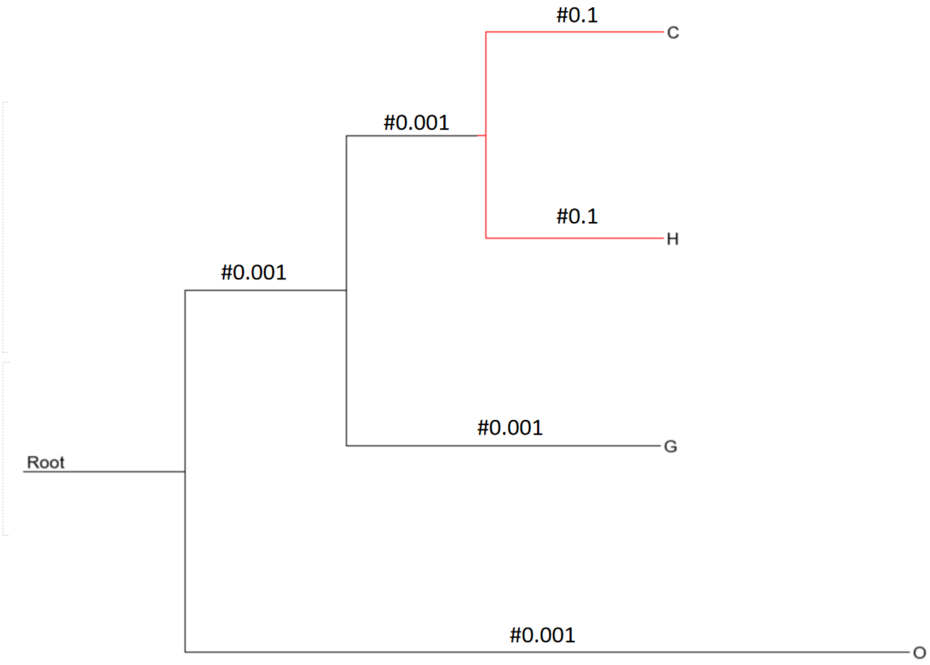


Figure 3.2: Visualization of species tree in simulation A.

The horizontal branch length in figure 3.2 represent time, the longer the branch, the more genetic changes has occurred. If we set all the effective population size parameter to 0.001. the sequences generated will be too easy to classify. We set a value of 0.1 for the red branch, making classification of sequences from H (Human) and C (Chimpanzee) more difficult than sequences from G (Gorilla) and O (Orangutan) (Orangutan).

3.3 Sim B

In simulation B, we used the following tree: $(((H : 0.00402 \ \#0.1, C : 0.00402 \ \#0.1) : 0.00304 \ \#0.01, G : 0.00707 \ \#0.01) : 0.00929 \ \#0.01, O : 0.01635 \ \#0.01) \ \#0.01$. As we mentioned earlier in simulation A, Values after "#" mutation rate multiplied by effective population size. Branch length represents evolutionary time. The branch length has not been changed in simulation B, while the value θ is set to 0.01 except for red branches. Thus, the genetic difference between H and C remains the same. However, because of the larger *theta* value along the blue branches in figure 3.3, it is more difficult to distinguish species G, O from species H, C. As a result, our data set is more difficult to classify than in simulation A.

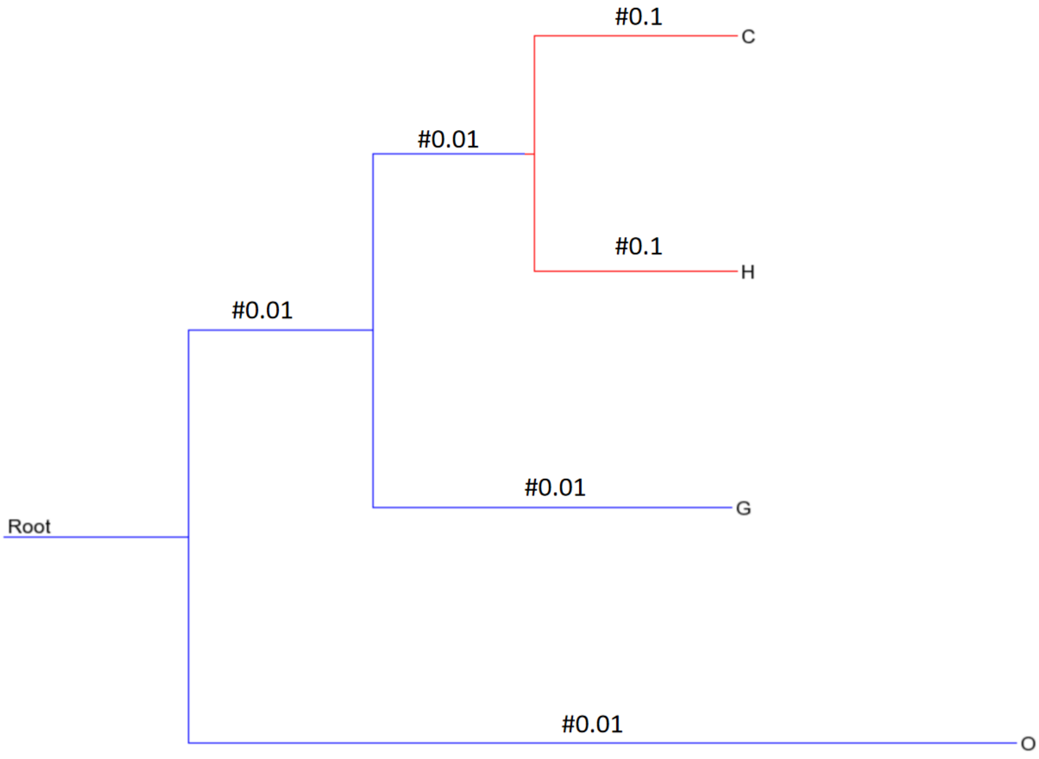


Figure 3.3: Visualization of species tree in simulation B.

3.4 Sim C

In simulation C, we used the tree $(((H : 0.00402 \ \#0.1, C : 0.00402 \ \#0.1) : 0.00304 \ \#0.1, G : 0.00707 \ \#0.1) : 0.00929 \ \#0.1, O : 0.01635 \ \#0.1) \ \#0.1$. This time we set all θ values to 0.1, which indicates a larger effective population size along all branches in figure 3.4. As a result, sequences generated by simulation C are the most difficult to analyze.

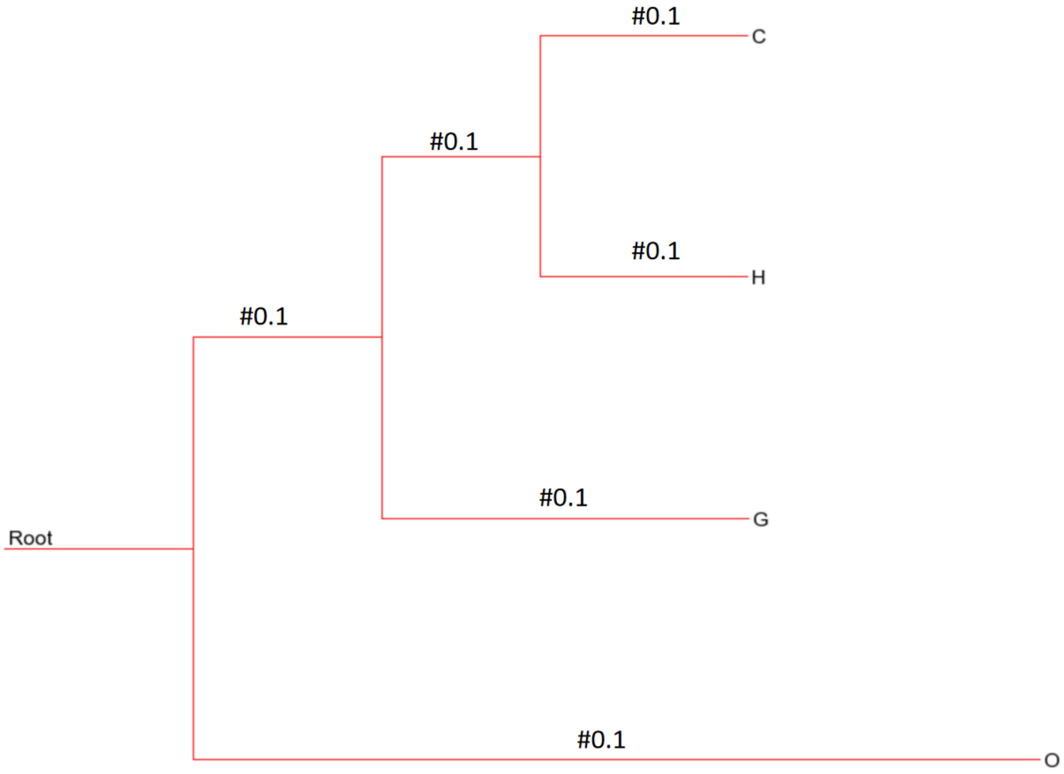


Figure 3.4: Visualization of species tree in simulation C and D.

3.5 Empirical data sets

3.5.1 Australian Sand dragon

In Australia, the newly arid habitat has led to severe ecological variation. As arid habitat expands, it was believed that the previously ecologically distinct lineages within the Australian sand dragon*Ctenophorus maculatus* species complex have diversified recently(Edwards et al., 2015). To study how species diversification is affected by ecological divergent evolution, multi-locus genetic data were collected from 153 individuals across 11 species within the *Ctenophorus maculatus* species complex to perform phylogenetic analysis (Edwards & Knowles, 2014). DNA sequence are available for two genes *ND2* and *16S*. Environmental and morphological data were also collected.

Genetic data of male and female samples were stored in different files in which each sequence has a unique label. The morphological data file contains species name information: its first column record the unique label and the second column is the respective species name. Hence, we have to change unique label to species name for each sequence. To reduce the workload, We choose 81 sequences out of 153 individuals to perform species classification.

Data from two gene locus for the 81 sequences were available. Then we concatenate sequences from two files according to their label. The sequence is already aligned, each ND2 locus DNA is about 1556 bases long, while 16s locus is 432 bases long, so each concatenated sequence is 1988 bases long. The sequence file was uploaded to Raxml (Stamatakis, 2014) web page to generate a phylogenetic tree. The tree was constructed based on GTR substitution model by using the maximum likelihood method. 11 species were allocated in 11 clades, indicating significant species boundaries among them.

There are unknown nucleotides in the sequence that were denoted by ”-”, we code this word as (0.5, 0.5, 0.5, 0.5). For this data set, ten-fold cross-validation is performed. The mean accuracy was computed.

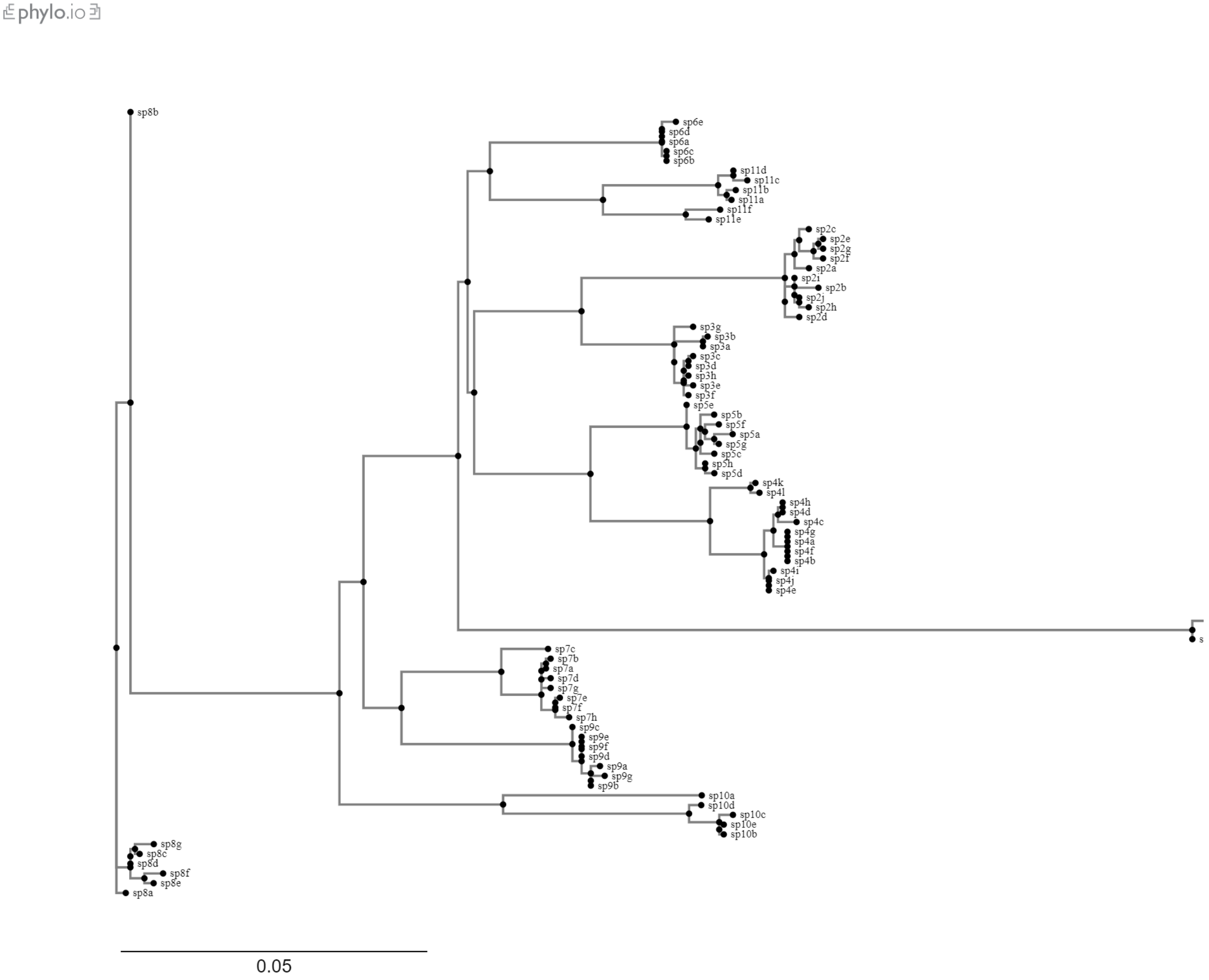


Figure 3.5: Phylogeny of sand dragon population.

3.5.2 Gekko

Lanyu Island is famous for its various endemic species that are shared with Asian mainland and northern Philippines island (Siler et al., 2012). A recent study has demonstrated that the rare endemic Orchid Island Gecko, *Gekko Kikuchii* share similar morphological characters with the Philippine species *G. mindorensis* (Siler et al., 2014). To perform a phylogenetic analysis, multilocus data were collected for 72 individuals across 12 species within the *Gekko* species complex (Siler et al., 2012). Sequence data were available for ND2 gene, as well as the nuclear gene *PDC*. Four additional nuclear genes *R35*, *L53*, *L78* , *L145* were also sequenced by Siler et al.(2014).

Sequence data for the above six genes were already concatenated and aligned. Each sequence is 3119 bases long. There were no missing data or gaps. The sequence file was uploaded to Raxml (Stamatakis, 2014) webpage to generate a phylogenetic tree. The tree was constructed based on GTR substitution model by using the maximum likelihood method. Figure 3.6 illustrates the phylogeny of our data, from which we can see two individuals of *Gekko Kikuchii* species were located in *G. mindorensis* clades, six individuals of another species were also misclassified into these two clades, indicating less genetic divergence between these three species.

There are ambiguous nucleotides in sequences denoted by the sign ”?”. To transform the sequence into a vector, we code this word as (0.5, 0.5, 0.5, 0.5). Ten-fold cross-validation for our neural network model and other machine learning approaches were performed on this data set. The mean accuracy was computed.

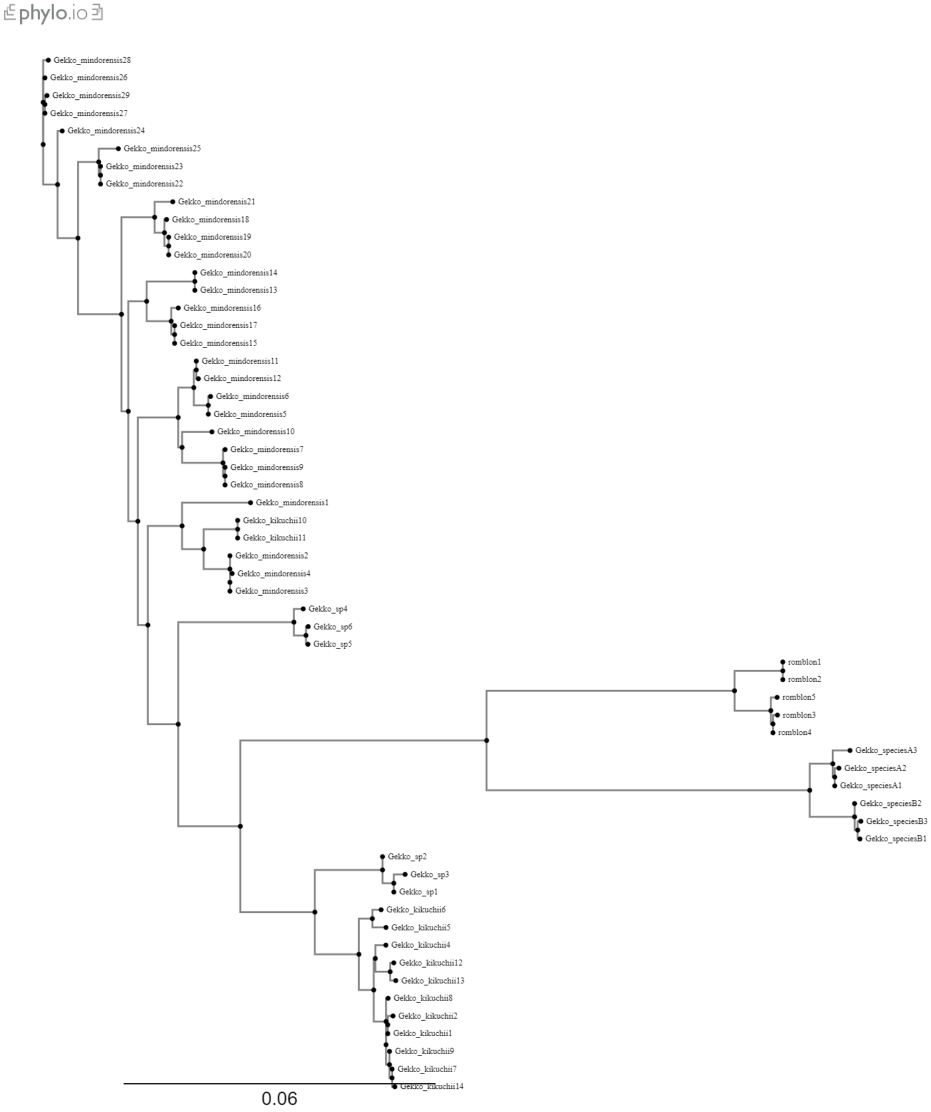


Figure 3.6: Phylogeny of Geeko population.

Chapter 4

Analysis and Results

4.1 Mathematical Statistics Tools

Generally, the performance of a model is measured by mean squared error, which evaluates how far the estimated value is from the true value. However, this statistic can not be utilized to evaluate the classification model. For machine learning classifiers, the accuracy is defined by:

$$accuracy = \frac{correctly\ classified\ sequences}{all\ sequences}$$

As we have mentioned in chapter 3, For each scenario we simulated 50 samples. Then the averaged accuracy of 50 repetitions was computed for each method. Let A_i denotes the accuracy for the i-th experiment, the average accuracy is computed by:

$$\overline{A_{50}} = \frac{1}{50}\sum_{i=1}^{50} A_i$$

Suppose the accuracy of an algorithm follows a discrete distribution with mean μ. According to the law of large number, estimated average accuracy $\overline{A_n}$ converges to the true mean μ as n goes to infinity. Our n=50 is large enough, average accuracy $\overline{A_{50}}$ should be close to mean μ. This is the main reason we repeat each experiment 50 times. The average accuracy $\overline{A_{50}}$ was used to estimate the true mean of an algorithm since it is an unbiased estimator.

$$E(\frac{1}{50}\sum_{i=1}^{50} A_i) = \frac{1}{50}\sum_{i=1}^{50} E(A_i) = \mu$$

Variance measures stability of algorithms. Let $Var(A_i) = \sigma^2$ denote variance of the method accuracy. Unbiased estimator of variance is given by: $\frac{1}{49}\sum_{i=1}^{50}\left(A_i - \overline{A_{50}}\right)^2$. For normal distribution, approximately 95 percent of data is within 2 standard deviations. According to central limit theorem, average sample accuracy $\frac{1}{n}\sum_{i=1}^{n} A_i$ follows normal distribution with mean μ and variance σ^2/n as n goes to infinity. Our n is large (>30), utilizing central limit theorem. We have:

$$P\left(\mu - 1.96\frac{\sigma}{\sqrt{n}} < \frac{1}{n}\sum_{i=1}^{n} A_i < \mu + 1.96\frac{\sigma}{\sqrt{n}}\right) = 0.95$$

Substitute μ and σ^2 with $\frac{1}{50}\sum_{i=1}^{50} A_i$ and $\frac{1}{49}\sum_{i=1}^{50}\left(A_i - \overline{A_{50}}\right)^2$ respectively, we get the 95 percent confidence interval.

4.2 Sim A

The accuracies on synthetic data sets of simulation A are summarized in table 4.1. The results suggest our neural network method reaches the highest classification. When sequences data are available for more than 10 genes, the mean accuracy is approximately equal to 1. The Figure 4.1 visualize accuracy table 4.1. The x-axes show the number of genes, y-axes represent accuracy. Different colors represent different methods. It is notable that SVM algorithm obtains a slightly larger accuracy than NN for single-locus data. The 95 percent confidence interval for NN and SVM were (0.7596017, 0.8243983) and (0.7904565, 0.8381149) respectively. These confidence intervals should be interpreted as: if 100 averaged values are calculated, then 95 of them will be located in the confidence level. Heavily overlapping confidence interval indicates two methods obtain close mean accuracy.

Table 4.1: Accuracies for synthetic datasets of simulation A[%].

#Genes	Seq.length	NN	Jrip	Naive Bayes	SVM	J48
1	250	79.23	62.85	71.78	81.42	76.78
2	500	85.05	66.36	72.27	79.54	76.81
4	1000	92.63	69.25	70.50	82.50	76.25
10	2500	98.34	74.79	70.58	95.21	83.12
20	5000	99.00	75.36	71.32	96.07	80.71

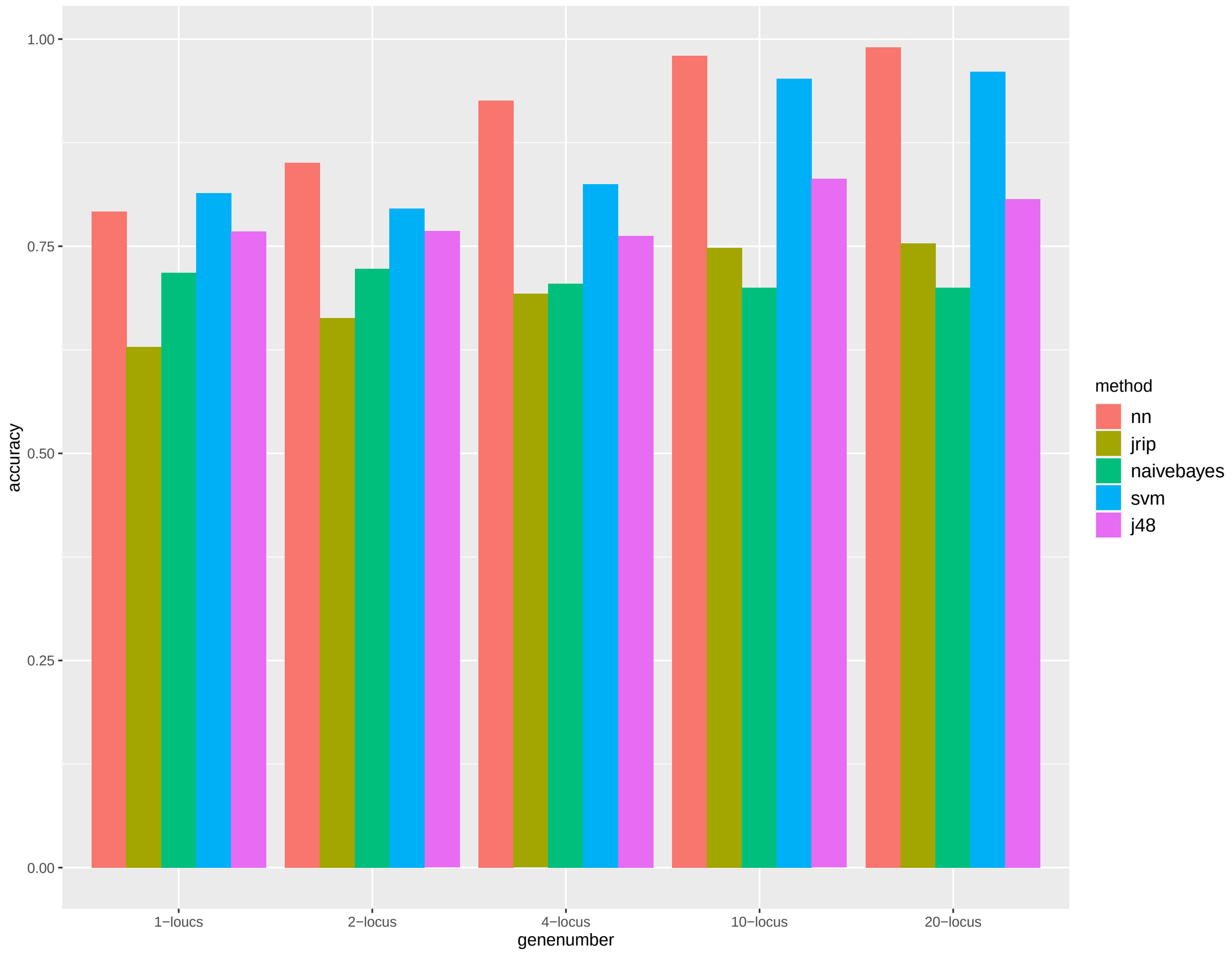


Figure 4.1: Barplot of accuracy table 4.1

Another statistic of interest is standard deviation, which measures the stability of classification performance. A large standard deviation value indicates that the method, despite having a high level of mean accuracy, may perform poorly sometimes. In practice, this can be disastrous: incorrect classification of invasive species can lead to the extinction of native animals and plants, causing ecological and environmental damage to the new era. Thus the standard deviation must be checked for our neural network model. Instead of plotting standard deviation data points, 5 confidence intervals are provided in Figure 4.2. The number of genes (x-axes) is denoted by N, the y-axes represent accuracy. The width of a confidence interval is proportional to its standard deviation. The figure provides an indication that increasing the number of genes significantly reduces the standard deviation, Thus, multi-locus data is critical not only for increasing the mean accuracy of our neural network classifiers, but also for reducing classification performance variation.

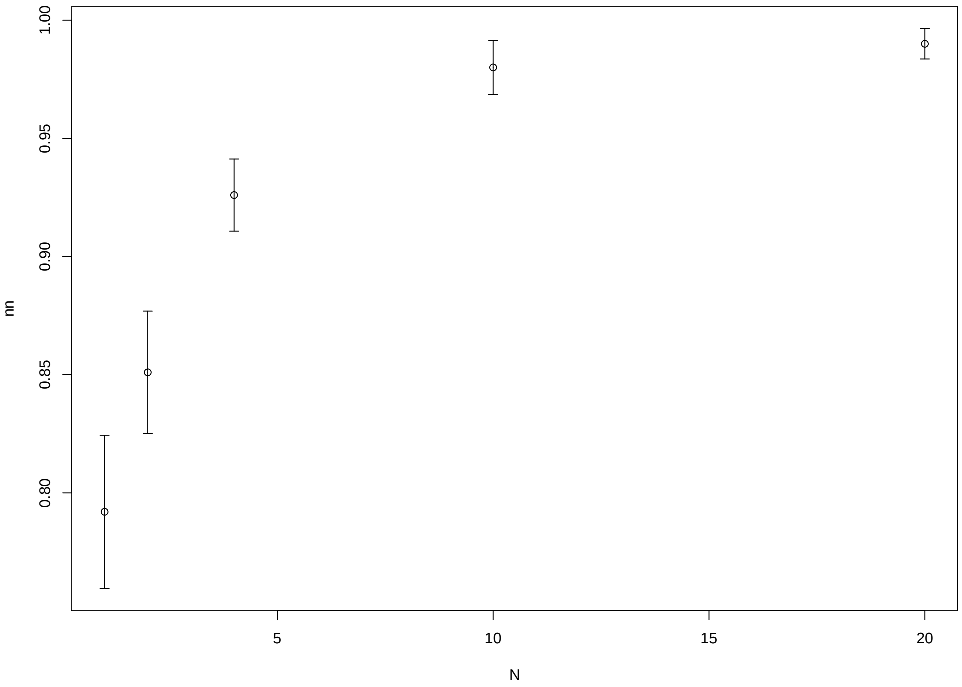


Figure 4.2: Confidence interval for neural network in SimA

4.3 Sim B

The classification results tested on sequences simulated from the first species tree in simulation A revealed that our neural networks method could achieve a higher level of mean accuracy than the other four methods. However, it appears that the previously simulated sequences were too easy for our neural network classifier. We generated sequences from another species tree to fully test its power. Before feeding these sequences to our model, the GMYC (Pons et al., 2006) model was used to assess the difficulty level of determining species boundaries. The results show that 100 sequences diverged into more than 15 species, indicating that species boundaries are fuzzy.

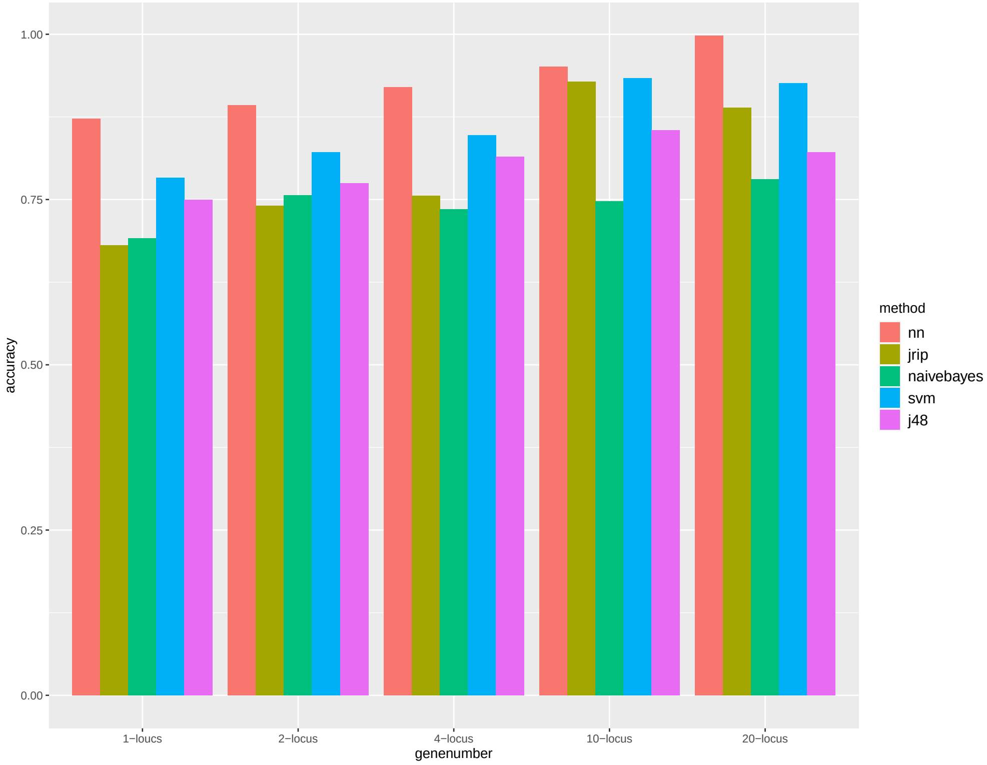


Figure 4.3: Barplot of accuracy table 4.2

The figure 4.3 depicts the mean accuracy table of Simulation B. The results of Simulation B are mostly consistent with the results of Simulation A: the neural network classifier achieves the highest accuracy, followed by the SVM classifier. Other classifiers are outperformed by neural network classifiers and SVM approaches. However, it appears that sequences simulated in simulation B are easier to classify because the mean accuracy of the neural network obtained in simulation B is slightly higher than that obtained in simulation A. This outcome contradicts our parameter settings for species tree B.

Table 4.2: Accuracies for synthetic datasets of simulation B [%].

#Genes	Seq.length	NN	Jrip	Naive Bayes	SVM	J48
1	250	87.21	68.06	69.17	78.33	75.00
2	500	89.32	74.06	75.62	82.19	77.50
4	1000	92.05	75.59	73.53	84.71	81.47
10	2500	95.18	92.86	74.76	93.33	85.48
20	5000	99.83	88.91	78.04	92.61	82.17

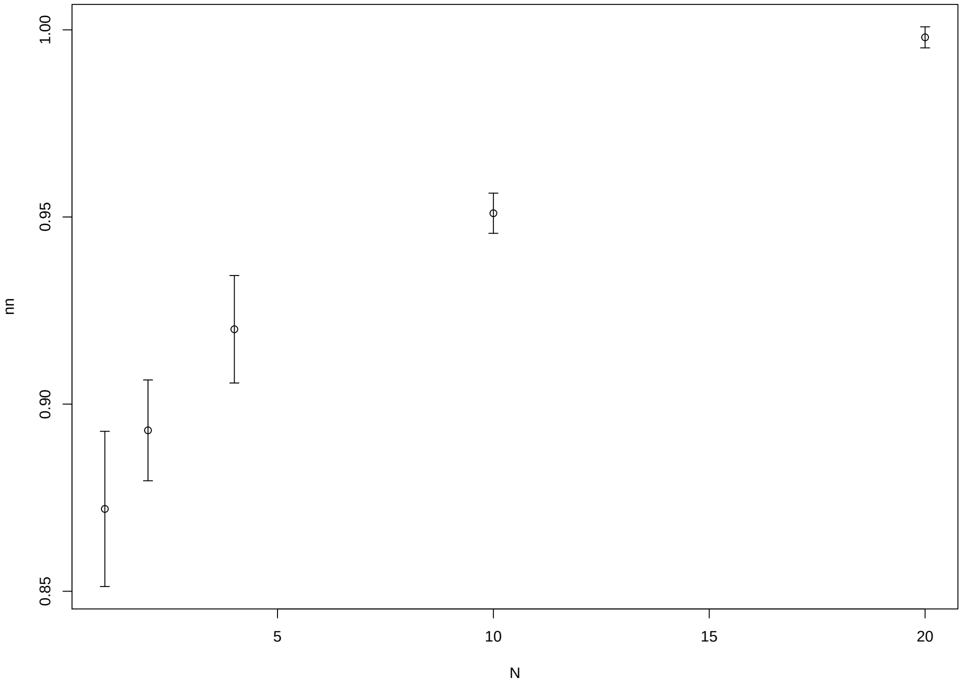


Figure 4.4: Confidence interval for neural network in SimB

In species tree B, we change parameters to make the simulated DNA sequences more difficult to classify. However, in simulation B, the NN classifier has slightly higher accuracy than in simulation A. This could be explained by the high standard deviation shown in 4.4 above. Because the standard deviation is large for 1-locus, 2-locus, and 4-locus data, we may obtain higher mean accuracy for the NN classifier in simulation B than in simulation A due to random error. The mean accuracy for the NN classifier in simulation B is close to that in simulation A for 10-locus and 20-locus data. We should increase our sample size to get a more accurate estimate of the mean accuracy.

4.4 Sim C

The DNA sequences generated from species tree C are the hardest to classify. Thus we obtain lower mean accuracies in simulation C than in simulation B(A). The results for simulation C are largely consistent with results of previous simulations: our neural network method continues to outperform the other four approaches in classification performance.

Table 4.3: Accuracies for synthetic datasets of simulation C [%].

#Genes	Seq.length	NN	Jrip	Naive Bayes	SVM	J48
1	250	85.37	48.00	35.00	78.50	52.00
2	500	90.75	56.54	60.00	72.31	62.31
4	1000	86.61	84.29	82.86	90.00	85.36
10	2500	92.82	75.00	75.62	90.31	78.75
20	5000	94.00	72.22	76.11	90.83	78.33

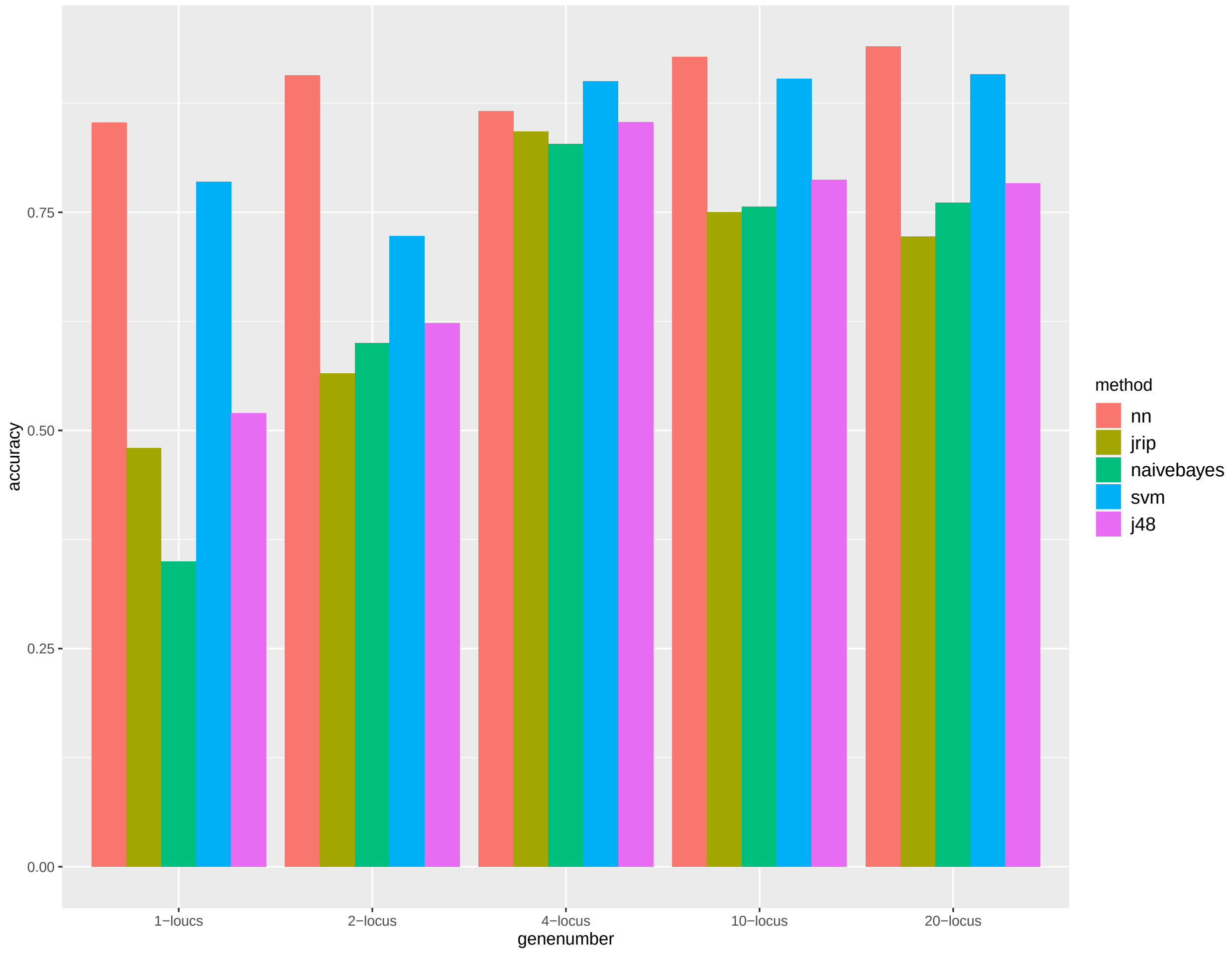


Figure 4.5: Barplot of accuracy table 4.3

4.5 Empirical Data Sets

The Australian sand dragon data set contains 81 sequences, each of which is made up of two periods taken from the textitND2 and textit16S genes. This data set contains two loci of DNA sequences, indicating a high standard deviation for classification accuracy. As a result, our neural network's classification performance is unstable, and we may obtain an accuracy that is significantly lower than the mean accuracy. This could explain why the SVM, J48, and Jrip methods perform better than our NN method.

DNA sequences for six genes are available for the Gekko data set. On 6-locus DNA sequence data, the performance of our neural network method is more stable. As a result, the obtained accuracy (98.33%) is close to the mean accuracy, indicating that our neural network method performs well on the Gekko data set.

Table 4.4: Accuracies on empirical data sets [%].

Species	seq.length	NN	Jrip	Naive Bayes	SVM	J48	GMYC
Sand Dragon	1988	90.61	92.45	32.21	92.95	95.77	82.71
Gekko	3119	98.33	96.61	62.71	96.35	92	81.35

The results of simulation and real-world data analyses show that neural networks outperform other approaches in a variety of scenarios. However, keep in mind that classification performance for 1-locus or 2-locus data can be unreliable. We should use data from multi-locus DNA sequences to determine species boundaries.

Chapter 5

Discussion

Our main goal is to select an approach to the problem of inferring species membership of an unknown specimen by analyzing its nucleotides sequence. Such a task was addressed using a neural network implemented by python software(Van Rossum & Drake, 2009). The classification accuracies were compared with respect to machine learning classifiers like Support Vector Machines (SVM), rule-based method RIPPER (Jrip), and the decision tree C4.5, Naive Bayes. Additionally, General Mixed Yule Coalescent (GMYC) was also tested on empirical data sets.

The classification analysis shows neural networks method is a promising tool in species delimitation. Higher classification performances on simulated data sets have been obtained compared to other methods. Compared to statistical model-based approaches, our neural network method is based on fewer or almost no assumptions when making predictions, whereas almost all model-based approaches rely on model assumptions that may not apply to real data (Zhang et al., 2008). Moreover, the machine learning method is computationally efficient, while model-based approaches usually require a long computational time and failed to handle a large amount of sequences data. However, model-based approaches such as GMYC do not require training data, so it is to some degree unfair to compare their accuracies with respect to machine learning methods. When there is no enough reference set available, statistical models are the best choice to infer species membership.

To make predictions, the neural network method has the potential to use other types of data. For example, morphological characters and DNA sequences can be coded to real-valued vectors. The neural network model is then trained and tested using these real-valued vectors. As a result, the classification results will be more accurate and reliable.

Even though we have demonstrated that the neural network method obtains the highest classification accuracy compared to other methods, we also note that neural network method is not without problems. First, the neural network method can only associate an unknown sequence with a known species label. The neural network method is inapplicable for the discovery of unknown species in this context. Second, we must train the neural network classifier with well-labeled sequences. However, training sequences are not always available.

With the rapid advancement of DNA sequencing techniques, it is now much cheaper and easier than ever before to obtain whole genome sequence data for many organisms of interest. Thus, if we need training sequences to infer the membership of an unknown sequence, we can use BLAST to find sequences from related species, then use these sequences to train our classifier and make predictions. Another approach is to reconstruct a phylogenetic tree of unknown sequences, after which we can simulate sequences from the phylogenetic tree using the procedure described in Chapter 3. These simulated sequences can be used to train our model.

Appendix A

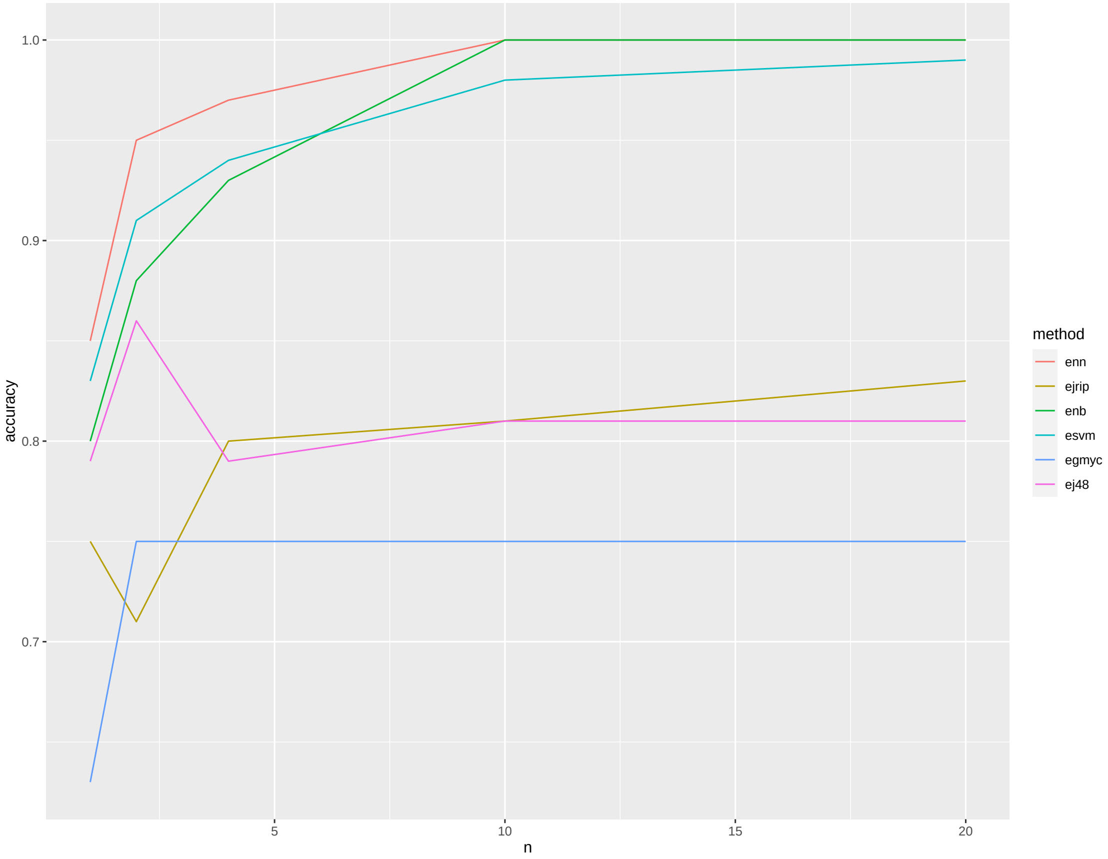


Figure A.1: Visualization of accuracy table 4.1

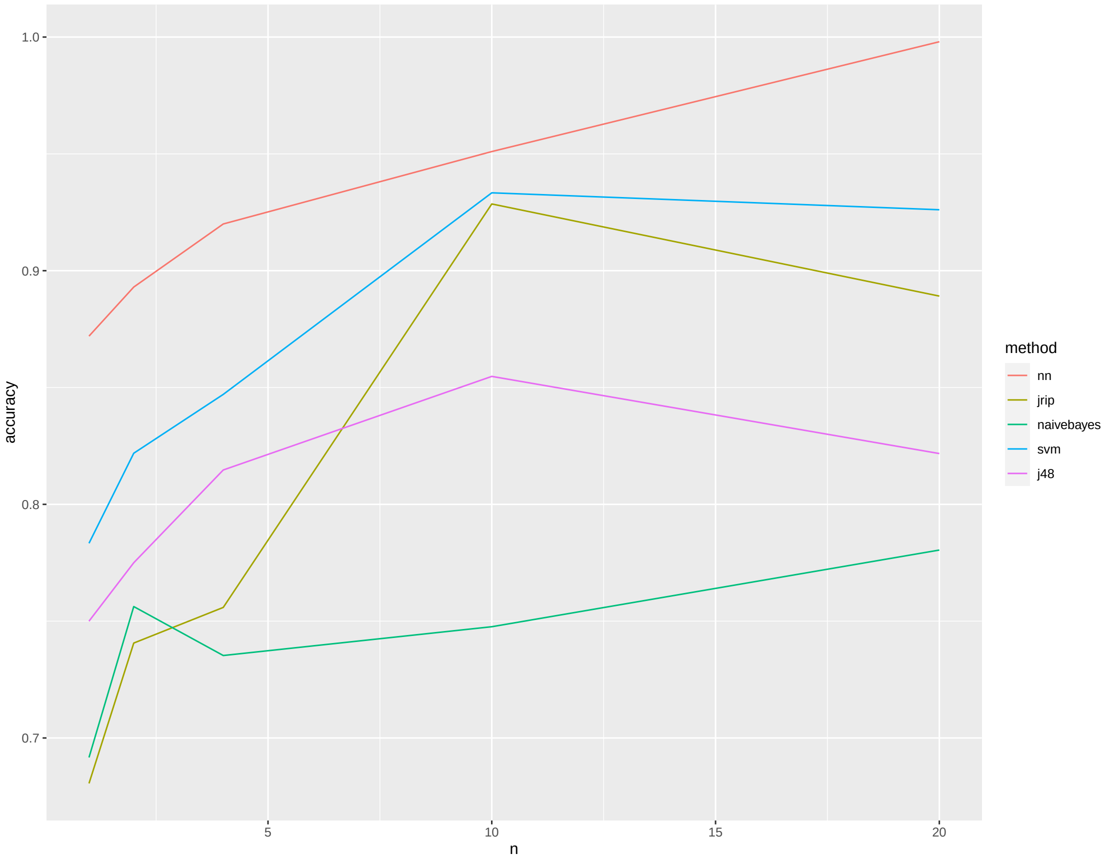


Figure A.2: Visualization of accuracy table 4.2

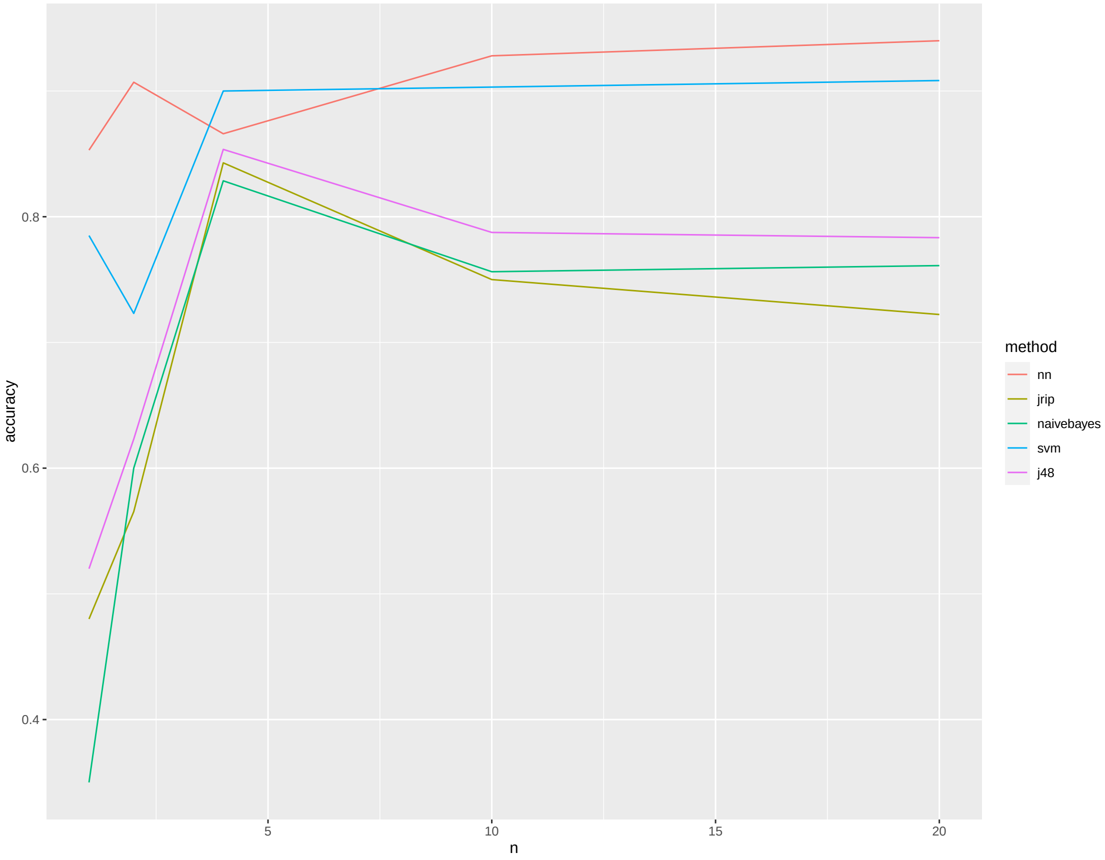


Figure A.3: Visualization of accuracy table 4.3

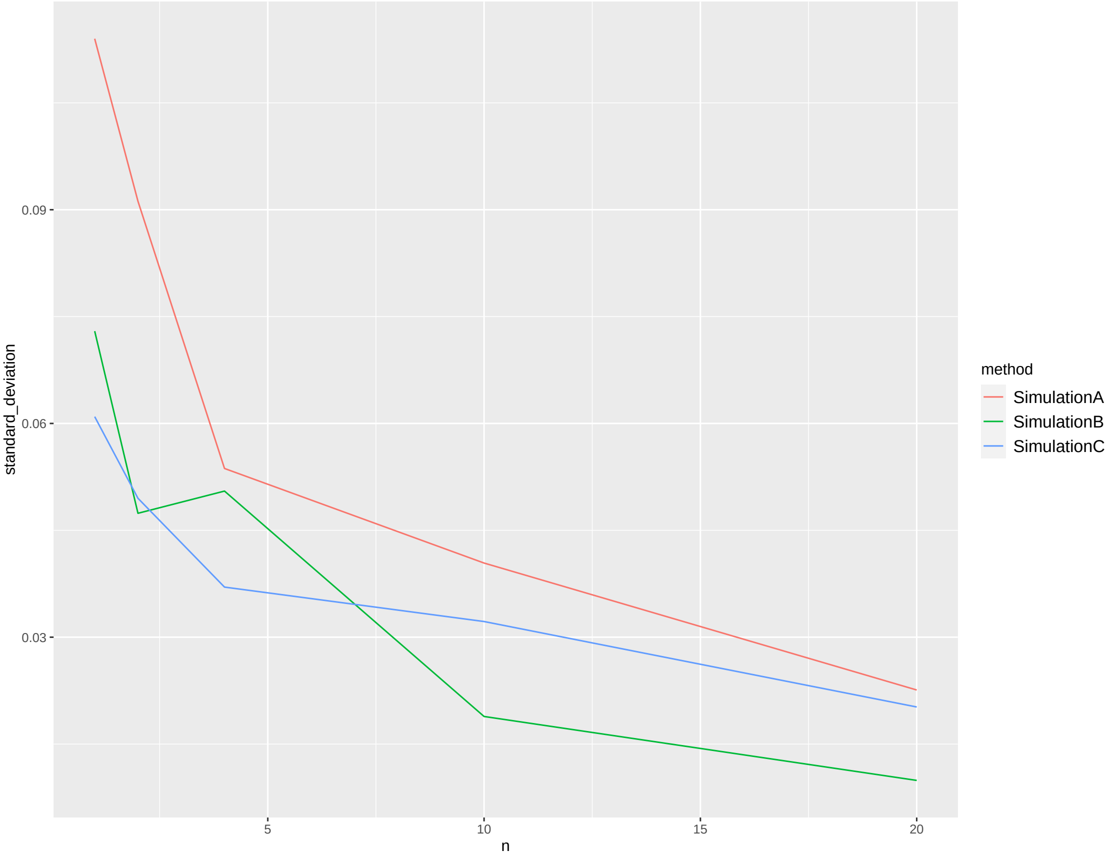


Figure A.4: Standard deviation plot for neural network classifier

Bibliography

Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, … Xiaoqiang Zheng. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems [Software available from tensorflow.org]. https://www.tensorflow.org/

Drummond, A. J., Ho, S. Y. W., Phillips, M. J., & Rambaut, A. (2006). Relaxed phylogenetics and dating with confidence. *Plos Biology*, *4*(5), 699–710. https://doi.org/ARTNe8810.1371/journal.pbio.0040088

Edwards, D. L., & Knowles, L. L. (2014). Species detection and individual assignment in species delimitation: Can integrative data increase efficacy? *Proceedings of the Royal Society B: Biological Sciences*, *281*(1777), 20132765. https://doi.org/10.1098/rspb.2013.2765

Edwards, D. L., Melville, J., Joseph, L., & Keogh, J. S. (2015). Ecological divergence, adaptive diversification, and the evolution of social signaling traits: An empirical study in arid australian lizards. *The American Naturalist*, *186*(6), E144–E161. https://doi.org/10.1086/683658

Esselstyn, J. A., Evans, B. J., Sedlock, J. L., Khan, F. A. A., & Heaney, L. R. (2012). Single-locus species delimitation: A test of the mixed yule-coalescent model, with an empirical application to philippine round-leaf bats. *Proceedings of the Royal Society B-Biological Sciences*, *279*(1743), 3678–3686. https://doi.org/10.1098/rspb.2012.0705

Ezard, T., Fujisawa, T., & Barraclough, T. (2009). Splits: Species' limits by threshold statistics. r package version 1.0-14/r31.

Fujisawa, T., & Barraclough, T. G. (2013). Delimiting species using single-locus data and the generalized mixed yule coalescent approach: A revised method and evaluation on simulated data sets. *Systematic Biology*, *62*(5), 707–724. https://doi.org/10.1093/sysbio/syt033

Hebert, P. D. N., Cywinska, A., Ball, S. L., & Dewaard, J. R. (2003). Biological identifications through dna barcodes. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, *270*(1512), 313–321. https://doi.org/10.1098/rspb.2002.2218

JUKES, T. H., & CANTOR, C. R. (1969). Chapter 24 - evolution of protein molecules. In H. MUNRO (Ed.), *Mammalian protein metabolism* (pp. 21–132). Academic Press. https://doi.org/https://doi.org/10.1016/B978-1-4832-3211-9.50009-7

Liu, L., & Yu, L. L. (2010). Phybase: An r package for species tree analysis. *Bioinformatics*, *26*(7), 962–963. https://doi.org/10.1093/bioinformatics/btq062

Lukhtanov, V. A. (2019). Species delimitation and analysis of cryptic species diversity in the xxi century. *Entomological Review*, *99*(4), 463–472. https://doi.org/10.1134/s0013873819040055

Luo, A., Ling, C., Ho, S. Y. W., & Zhu, C. D. (2018). Comparison of methods for molecular species delimitation across a range of speciation scenarios. *Systematic Biology*, *67*(5), 830–846. https://doi.org/10.1093/sysbio/syy011

Meyer, C. P., & Paulay, G. (2005). Dna barcoding: Error rates based on comprehensive sampling. *PLOS Biology*, *3*(12), e422. https://doi.org/10.1371/journal.pbio.0030422

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Platt, J. (1999). Fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods: Support Vector Learning*, 185–208.

Pons, J., Barraclough, T. G., Gomez-Zurita, J., Cardoso, A., Duran, D. P., Hazell, S., Kamoun, S., Sumlin, W. D., & Vogler, A. P. (2006). Sequence-based species delimitation for the dna taxonomy of undescribed insects. *Systematic Biology*, *55*(4), 595–609. https://doi.org/10.1080/10635150600852011

Puillandre, N., Lambert, A., Brouillet, S., & Achaz, G. (2012). Abgd, automatic barcode gap discovery for primary species delimitation. *Molecular Ecology*, *21*(8), 1864–1877. https://doi.org/10.1111/j.1365-294x.2011.05239.x

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers Inc. http://portal.acm.org/citation.cfm?id=152181

Rambaut, A., & Grassly, N. C. (1997). Seq-gen: An application for the monte carlo simulation of dna sequence evolution along phylogenetic frees. *Computer Applications in the Biosciences*, *13*(3), 235–238. %3CGo%20to%20ISI%3E://WOS:A1997XC29200005

Rannala, B., & Yang, Z. H. (2003). Bayes estimation of species divergence times and ancestral population sizes using dna sequences from multiple loci. *Genetics*, *164*(4), 1645–1656. %3CGo%20to%20ISI%3E://WOS:000185248000036

Salzberg, S. L. (1994). C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning*, *16*(3), 235–240. https://doi.org/10.1007/bf00993309

Siler, C. D., Oaks, J. R., Cobb, K., Ota, H., & Brown, R. M. (2014). Critically endangered island endemic or peripheral population of a widespread species? conservation genetics of kikuchi's gecko and the global challenge of protecting peripheral oceanic island endemic vertebrates. *Diversity and Distributions*, *20*(7), 756–772. https://doi.org/10.1111/ddi.12169

Siler, C. D., Swab, J. C., Oliveros, C. H., Diesmos, A. C., Averia, L., Alcala, A. C., & Brown, R. M. (2012). Amphibians and reptiles, romblon island group, central philippines: Comprehensive herpetofaunal inventory. *Check List*, *8*(3), 443. https://doi.org/10.15560/8.3.443

Stamatakis, A. (2014). Raxml version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, *30*(9), 1312–1313. https://doi.org/10.1093/bioinformatics/btu033

Suchard, M. A., Lemey, P., Baele, G., Ayres, D. L., Drummond, A. J., & Rambaut, A. (2018). Bayesian phylogenetic and phylodynamic data integration using beast 1.10. *Virus Evolution*, *4*(1). https://doi.org/ARTNvey01610.1093/ve/vey016

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace.

Wang, Q., Garrity, G. M., Tiedje, J. M., & Cole, J. R. (2007). Naive bayesian classifier for rapid assignment of rrna sequences into the new bacterial taxonomy. *Applied and Environmental Microbiology*, *73*(16), 5261–5267. https://doi.org/10.1128/Aem.00062-07

Weitschek, E., Fiscon, G., & Felici, G. (2014). Supervised dna barcodes species classification: Analysis, comparisons and results. *Biodata Mining*, *7*. https://doi.org/10.1186/1756-0381-7-4

Wiemers, M., & Fiedler, K. (2007). Does the dna barcoding gap exist? – a case study in blue butterflies (lepidoptera: Lycaenidae). *Frontiers in Zoology*, *4*(1), 8. https://doi.org/10.1186/1742-9994-4-8

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques* (2nd). Morgan Kaufmann.

Yang, Z. (2015). The bpp program for species tree estimation and species delimitation.

Yang, Z., & Rannala, B. (2010). Bayesian species delimitation using multilocus sequence data. *Proceedings of the National Academy of Sciences*, *107*(20), 9264–9269. https://doi.org/10.1073/pnas.0913022107

Yang, Z., & Rannala, B. (2014). Unguided species delimitation using dna sequence data from multiple loci. *Molecular Biology and Evolution*, *31*(12), 3125–3135. https://doi.org/10.1093/molbev/msu279

Zhang, A. B., Sikes, D. S., Muster, C., & Li, S. Q. (2008). Inferring species membership using dna sequences with back-propagation neural networks. *Systematic Biology*, *57*(2), 202–215. https://doi.org/10.1080/10635150802032982