ANALYZING THE PERFORMANCE OF MACHINE LEARNING ALGORITHMS ON METAGENOMIC DATA

by

VASIM MAHAMUDA

(Under the Direction of Khaled Rasheed)

ABSTRACT

Metagenomics is a branch of bioinformatics that deals with the study and analysis of micro-organisms in natural environments. Some micro-organisms including many species of bacteria, archea and viruses should be studied in their natural habitat as these organisms cannot be cultivated in the laboratory by using standard techniques. Machine learning techniques are being applied to this field to predict novel genes. In this thesis, we try to address the issue of classifying metagenomic sequences. First, we compare the performance of several machine learning approaches including ensemble learners to identify which algorithms will be able to bin metagenomic data into taxa-specific bins with high accuracy. Then we do scalability studies to investigate how the performance of those algorithms degrades as the number of species in the metagenomic sample increases. We also study the performance degradation with the increase in the number of unknown sequences in the data. The results are very promising and show that machine learning algorithms perform very well in this domain. Futhermore, the performance degrades gracefully with the increase in the number of species and the number of unknown sequences.

INDEX WORDS: Binning, decision trees, machine learning, metagenomics, ensemble methods, supervised learning.

ANALYZING THE PERFORMANCE OF MACHINE LEARNING ALGORITHMS ON METAGENOMIC DATA

by

VASIM MAHAMUDA

B.Tech, Andhra University, India, 2007.

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

2010

ANALYZING THE PERFORMANCE OF MACHINE LEARNING ALGORITHMS ON METAGENOMIC DATA

by

VASIM MAHAMUDA

Major Professor:	Khaled Rasheed
Committee:	Hamid R. Arabnia Walter D. Potter

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
August 2010

DEDICATION

To my loving parents.

ACKNOWLEDGEMENTS

I would like to thank Dr. Rasheed, my advisor for his support, co-operation, and guidance from the time I was struggling to decide on a topic till the completion of my thesis. His insightful thoughts and advice led me to improve the work considerably and helped me finish my thesis in time. I would like to thank Dr. Arabnia and Dr. Potter for being part of my advisory committee. I am thankful to my research team member Manchon U. Initial help from Manchon U is really praise worthy. I would like to thank Dr. Katharina Hoff, for her time and effort. Dr. Hoff provided us with the background knowledge necessary for applying computational skills to metagenomes. I would also like to thank Eric Talevich for his help with some biological details. Also, I would like to thank Dr. Jan Mrazek for his suggestions and insightful thoughts which helped to improve the second article. I would also like to thank my fellow graduate students for their help and support. Finally, I would like to thank my parents, brothers, and sister-in-law for their unconditional love, guidance, moral support, and instilling in me the confidence which helped me in pursuing higher studies.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION AND LITERATURE REVIEW

Metagenomics is a new field of genomics, which deals with the study of uncultured organisms. Bacteria and archea are organisms that belong to a kingdom known as prokaryotes. Many of these organisms cannot be cultured in laboratory and hence they should be studied in their natural habitat [1]. The DNA of such organisms is obtained by using high throughput sequencing of environmental samples such as sea water, land and human guts.

Microbes are micro-organisms which are found everywhere ranging from sea water samples of the Sargasso Sea to the human body. Surprisingly, these organisms/prokaryotes are also present in human beings. A gut of the human being may have thousands of such microbes and they are responsible for the production of vitamin K in humans. So as to facilitate the study of these micro-organisms which are uncultivable, a new field has emerged in the area of genetic research known as 'metagenomics' [2].

The sequences which are produced from sequencing techniques are very short in length. The main concern which arises is that there is so much similarity between the sequence reads of organisms that biologists cannot assure that the fragments of sequences obtained belong to one particular species. The sequencing of these organisms to form a complete genome is also another challenging task. So far there are very few prokaryotic organisms whose genomes have been completely sequenced. Recently there has been work going on in this field to classify the organisms to their respective taxa specific bins. Sequence similarity based and sequence composition based techniques allow us to study the phylogenetic evolution of assembled contigs [3]. Sequence similarity methods such as BLAST [4] , MEGAN [5] are used to assign contigs based on some similarity measure. Clustering methods such as K nearest neighbor can be used to group the sequences into taxa specific bins.

The role of machine learning comes into picture when we have such problems of classification based on some previous knowledge or when there are few examples to learn from. There are many statistical methods which can easily generate models for such problems. When the classifier is given a set of examples with labels, called the ‘training data’ it builds a model on those examples. It also teaches the model how to behave when presented with examples the classifier has never seen before. The classifier accuracy is measured on how well this classifier performs on examples which it has never seen before, also known as the ‘test set’.

In this work we try to address the issue of binning the sequences into their respective species. We apply machine learning algorithms to metagenomic data to see with what accuracy the classifier can perform. We have three dimensions to the problem of classification/binning:

1. To see which machine learning algorithm performs well in classifying metagenomic data.

2. To see how the performance drops when the number of species in the sample increase from 15 to 300.
3. To see how the performance of the classifier changes as the percentage of unknown sequences increases from 15-90%.

In Chapter 2, we present our work to classify the sequence reads into respective species. We compared the performance of different classifiers, both supervised and meta-learners to see which classifier will be good at classifying sequence reads of metagenomic data. We used six supervised learners namely decision trees, Naïve Bayes, support vector machines, artificial neural networks, Bayesian networks and decision tables. We also analyzed the performance of meta-learners such as bagging, boosting and stacking.

The second part of this work is to see how the performance of the classifiers drops as the number of species in the metagenomic sample increases. We address this problem in Chapter 3. We also try to simulate a real world problem of mixed sequence reads in which sequences from numerous unknown species exist in the sample. We attempt to evaluate the methods to see how well the algorithms perform on such mixed data.

In short in Chapter 2, we applied different classifiers to simulated/synthetic metagenomic data to compare their performance and identify the best whereas in Chapter 3, we studied scalability in terms of both the number of species and also in terms of varying the number of unknown sequences in the sample.

CHAPTER 2

APPLICATION OF MACHINE LEARNING ALGORITHMS FOR BINNING METAGENOMIC DATA[1]

[1] Vasim Mahamuda, Manchon U, Khaled Rasheed, Accepted by the International Conference on Bioinformatics and Computational Biology (BIOCOMP 2010). Reprinted here with permission of publisher, 7/14/2010.

2.0 ABSTRACT

Machine learning algorithms are extensively being used in the field of bioinformatics. With the invent of new genome sequencing techniques there has been a considerable increase in the amount of data available in the biological databases. In research projects, such as determining the structure and function of biological molecules, the role of machine learning is evident. The machine learning algorithm learns a new concept or function from some past experience. When given new examples, it classifies them based on the concept learned. In this article, our goal is to empirically evaluate how well the standard machine learning algorithms perform in classifying metagenomic data. Our approach involves classifying the sequence reads into respective species by performing a codon analysis on the DNA sequences and extracting features that are representative of the sequence reads. We compared the performance of six well known supervised learners - decision trees, Naïve Bayes, support vector machines, artificial neural networks, Bayesian networks and decision tables . We also analyzed the performance of three different meta-learners namely bagging, boosting and stacking. Experimental results are described which investigate the technique being presented.

Keywords: Bioinformatics, binning, ensemble methods, machine learning, metagenomics, supervised learners.

2.1 INTRODUCTION

2.1.1 Overview

Metagenomics is the study of organisms that cannot be cultured in the laboratory. Metagenomic DNA sequences can be found in samples directly extracted from natural habitats such as land, sea water etc. Metagenomics facilitates the study of a large population of microbial organisms such as bacteria, archea, and viruses [1]. Microbial organisms can be found everywhere such as land, sea water and even in the gut of human beings. Extracting the genomic data from such environmental samples enables us to study all the micro-organisms ranging from bacteria, archea and microeukaryotes that are involved in regulating the earth's ecological balance. Bacteria, archea and viruses belong to the group known as prokaryotes - i.e. the organisms which lack a nucleus. The sequences which are directly taken from natural habitats such as land, sea water, or the gut of human beings are known as metagenomes and the study of this sequence data is known as metagenomics [2]. The sequencing of metagenomic data allows us to explore and analyze the organisms which cannot be cultured in the laboratory [1].

Machine learning algorithms are being applied to various application domains that are related to the field of bioinformatics. One such field is metagenomics. The role of bioinformatics in the field of metagenomics is: (1) to find genes for detecting novel proteins; (2) to find evolutionary relationships among organisms. It is obvious that these algorithms are being applied to sequence reads because the data set is large enough and finding a method which is capable of handling such huge amounts of data is required. At the same time the need for computational methods arise because of the fact that they automate or speed up the process. Machine learning is nothing but building a model based on certain data or previous knowledge, which is known as the training data. When we have an unknown/unseen example and want to predict which category/class it belongs to, we do so by using the model which has been built on some previous knowledge.

Consider '*n*' examples of the form $\{(x_1,y_1), \dots , (x_n,y_n)\}$. The function $y = f(x)$ varies for different inputs of '*x*'. The x_i values are vectors of the form $\{x_{i1}, x_{i2} , . . . , x_{in}\}$. These vectors are called features or attributes of x_i. These features can be either discrete valued or continuous. Y_i is known as the target attribute. The target attribute, also called label, can take either continuous or discrete values. If the target attribute takes values from a discrete set of classes {1, …, K}, then it is a classification problem. In case of a regression problem the target attribute will take values from the real line (continuous) [3]. Machine learning algorithms can be applied either to the problem of regression or classification. Regression is used for predicting the values of a continuous variable, whereas classification is used when we classify the examples into a number of discrete categories or classes.

When the learning algorithm is given '*n*' training examples of labeled data it learns the concept and outputs a model. When new/unseen examples, called the test set, are presented to the learned model it outputs a prediction of the target attribute value based on the concept learned. This is known as supervised learning. Some of the supervised learners are decision trees, support vector machines, artificial neural networks. On the other hand unsupervised learning, i.e. learning without a target attribute, usually involves similarity-based approaches. The examples are partitioned into different clusters or classes. When a new example is presented it is assigned one of these clusters based on the similarity. One of the well-known unsupervised clustering algorithms is K-means.

2.1.2 Related work

Metagenomics is an active area of research which deals with the study of the microbial world. It has been the major area of focus of many recent sequencing projects. Using metagenomics researchers were able to determine the functional role of the molecules and also the chemical reactions taking place which aid in the process of symbiosis. The researchers also focused on the phylogenetic classification of these organisms. Phylogenetic classification helped researchers to determine what other kind of species, genera or phyla are present in the metagenomic sample [4]. The sequence data which is extracted by shotgun sequencing or other functional sequencing methods yields sequences which are fragmented in nature. The reads obtained can belong to either different species or they can belong to the same species with different strains. Gene assembly of reads which belong to different species can be easily done, but assembling genes of same species with different strains is quite a challenging task [5].

The sequence data obtained from such sequencing methods highly depends on the environment from which the sample is taken. In other words, there is a high significant correlation between the sequencing data obtained and the origin of the sample [2]. There are a variety of genome assemblers available such as Arachne [6], PCAP [7], Atlas [8]. All the above mentioned genome assemblers work on the same principal concept of assembling genome sequences from shotgun reads. The assembled fragments are then searched for probable genes. The two famous gene-finding techniques are GLIMMER [9], FgenesB_Annotator [10]. Glimmer is a system for finding genes of prokaryotes such as bacteria, archae and viruses. It uses hidden markov models which are among the most popular statistical models. FgenesB_Annotator is a package which does an automatic annotation of bacterial genomes.

Assigning reads to reference genomes or finding the similarity to known species is termed as grouping or classifying the sequences. Researchers in the metagenomic community term it as “binning” [11]. Using this concept of binning, the reads are assigned their taxa and hence can be classified. There are both supervised and unsupervised learning mechanisms to find genes in reads of sequences or to assign them to reference genomes. In the case of supervised learning the reads should be assigned labels or target attribute values of reference genomes.

There are many different methods by which genes are being discovered. One common way is to take the sequence reads and perform a blast against the known genes. Some of the common supervised programs are BLAST [12] and MEGAN [13]. BLAST is a program which enables comparison of a query sequence with a database of known sequences. It then identifies the sequences which match the query sequence based on some statistical similarity measure of matches. Metagenomic data present challenges here, as there is little knowledge about these organisms and the fragments of sequences are thought to be incomplete. So even with BLAST we cannot obtain accurate similarity searches of the query sequence to the sequences present in the databases. MEGAN is another similarity search based tool. It offers both a graphical and statistical analysis of the sequence. The sequence is assigned the taxa based on sequence alignment. MEGAN assigns the new query sequence to the sequence which has the highest similarity score. Similarity based approaches such as BLAST and MEGAN can be of little help in finding novel genes when we consider metagenomic data as there could be no reference homologues present in the databases [2].

Another way to predict genes is to find them based on the structural analysis of the composition of the DNA sequence. This can be achieved by either finding the coding regions or the non-coding regions or just searching for ORFs in the sequence and then using these features to build statistical models which can predict novel genes. When we look at a DNA sequence we cannot find any difference between the metagenomic data and the data that is used in other genomic projects. To us it may seem as the sequence data of metagenomic samples is similar to that of other sequencing data because at the abstract level we know that sequences are nothing but strings of A, T, G and C. But if we take into consideration patterns, the biological structure, functional role and long range sequences, they differ [14].

Many programs have been developed for predicting genes, which make use of statistical models. Orphelia [15] is one such program which uses a two stage machine learning approach to compute the gene probability. The program uses artificial neural networks to find the probable genes in the given sequences. Orphelia has two different programs which work for sequence lengths of 300 bp and 700 bp. MetaGene [16] is another program which is used for sequence lengths of (~700 bp) and is used to predict genes in prokaryotes. MetaGeneAnnotator [17] is the successor of MetaGene, and is available as a web server application. GeneMark [18] is yet another program which uses unsupervised learning and the algorithm used is iterative hidden Markov models. The program is based on heuristic approaches. Even though all the above programs are available as a web server application MetaGene, MetaGeneAnnotator and GeneMark support only sequence reads which are of (~700 bp) in length where as Orphelia supports sequences of length 300 bp and 700 bp. Our approach supports variable length sequence reads.

The approach used in this article is different from the above mentioned programs as we are trying to classify the sequences into a known set of species rather than trying to find genes in a particular sequence. Our main goal is to assist in the process of finding relationships among species as to how similar they are. This is a novel approach to the problem of classification. This is accomplished by first finding the important attributes present in the sequence data. Then, these features are given as input to a machine learning algorithm. The algorithm learns from the data and presents a model to the user. The model is then validated for correctness, by giving it a set of examples known as the test set, to see how well the model performs on unseen examples. In [11], the authors used Naïve Bayes classifier for assigning reads to respective phylogenetic groups. In our study we have not limited ourselves to just one classifier but tried a variety of different classifiers, both supervised and ensemble learners to determine which classifier performs well on such metagenomic data.

The rest of the article is organized as follows: in Section 2.2 we introduce our methodology and show how the features are extracted from the sequence data. In Section 2.3 we present the performance of various algorithms. In Section 2.4 we discuss future work and the article is concluded in Section 2.5.

2.2 APPROACH

2.2.1 Data selection

The data which is used in our experiments is available on Comprehensive Microbial Resource [19]. The data is selected in such a way that each species selected has a different genus. We randomly select 75 different species based on the criterion that each species should belong to a different genus. This set of 75 species is equally divided into 3 groups of 25 species each. From each of these groups we remove 10 species to form the set of 15 species. Two sets of experiments are performed for each group: one with the initial set of 25 species and the other with the set of 15 species. As the species selection is stochastic, in order to validate our approach we report the average accuracy of the 3 groups. As shown in Fig. 2.1, species which are selected for our experiments belong to different genus.

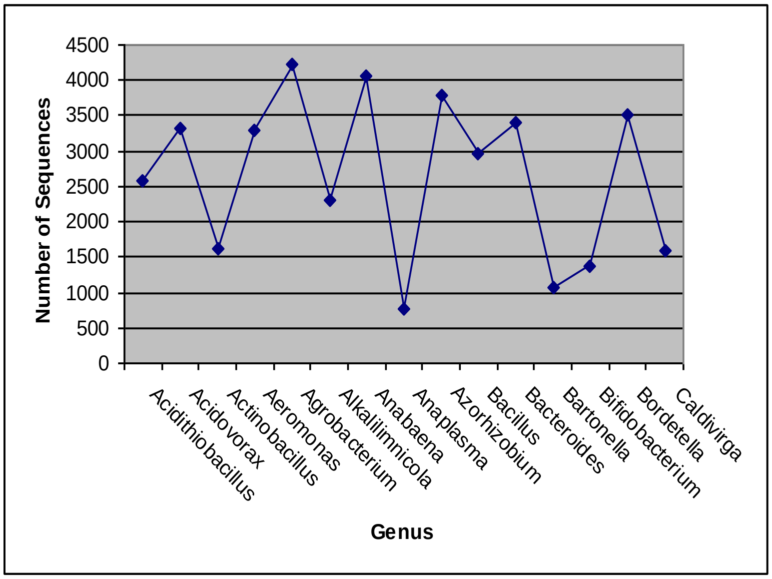

Figure 2.1 Shows how sequence lengths are distributed among the different genera.

2.2.2 Data pre-processing

The FASTA file of each species contains a number of different reads of DNA sequences. The number of reads and the average length of each read vary according to the species being selected. The average number of reads of DNA sequences in a file of each species would be approximately 3500 and the length of DNA sequence would be between 300-1000 bp.

The next step in our approach is to extract the features or attributes, for the machine learning algorithms to learn the concept and then to bin them into groups. The attributes or features we selected are the GC content, number of ORFs, uni-base frequency, di-base frequency, average length of ORF. GC content is the most important feature to consider because of the relative stability of the bond between the G and C bases rather than A and T bases [20]. If the DNA sequence has a pattern such that it begins with a start codon (ATG,CTG,GTG, or TTG), which is followed by at least 54 bp, and then ends with a stop codon (TGA,TAG, or TAA) then such a pattern is known as an 'Open Reading Frame or ORF [15]. Codons in biology are subsequent, non-overlapping triplets. We considered ORFs with overlap as the relative lengths of the sequences are short. By considering ORFs with overlap we get a good count of the number of ORFs in a sequence even though the sequences are short [15].

The DNA sequences which are in the FASTA format have a sequence identifier followed by the sequence itself. We first compute the GC content and then find the probable ORFs for each read of DNA sequence. As the sequence can be translated in six different ways, we find the ORFs in all possible frames (3 on positive strand, 3 on the minus strand). Now iterating through each ORF we look for codons with a pattern. Uni-base frequency is the term, which we use for codons which contain only one unique nucleotide, namely AAA, TTT, GGG, CCC. Di-base frequency is the frequency of the codons which contain two unique nucleotides, such as ATA, TTA, GGC, CGC and so on. We did not consider the feature tri-base frequency (codons which differ by all three nucleotides) as the features uni-base, di-base and tri-base are complementary.

This set of five features along with the species label (i.e. to which species the sequence belongs) forms the input to the machine learning algorithm. One sample tuple would be of the form (number of ORFs, uni-base frequency, di-base frequency, average length of ORFs, GC content, name of the species). Here, the name of the species is the target attribute which is discrete valued. In this way we process all the sequences such that for each sequence we have a tuple which gives a statistical representation of the underlying data. Fig. 2.2 shows a diagrammatic representation.

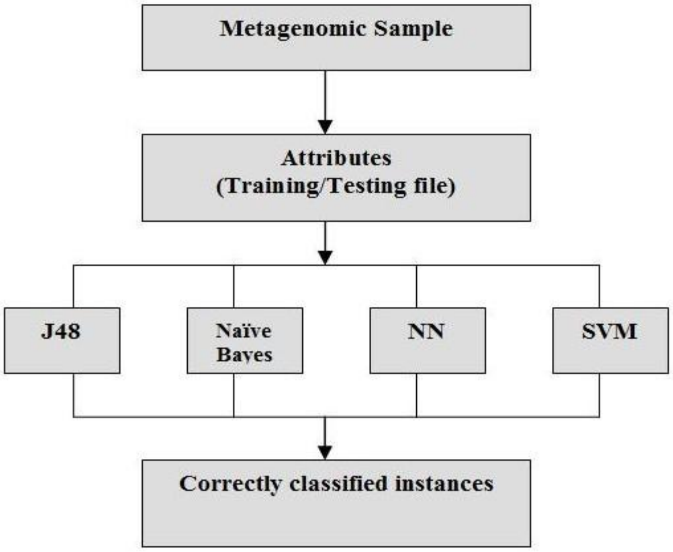

Figure 2.2 Step-by-step illustration of our approach

After the data has been processed and the features extracted the next step is to present it as an input to some machine learning algorithm.

2.2.3 Learning methods

The machine learning algorithms which we use are decision trees, decision tables, neural networks, support vector machines, and Naïve Bayes, Bayesian networks. Ensemble learners such as bagging, boosting and stacking are also used. We use the University of Waikato's WEKA [21] software which is an open source data mining software written in Java, to run the experiments. The Weka package has implementations of various popular machine learning algorithms.

We use a validation method approach wherein the data is randomly split into two sets of 80% and 20% known as the training/validation set and the testing set. The algorithm builds the model based on the training set and then the test set is presented to the model to see how well it performs. Another way of validating the data is by using N -fold cross validation approach. In this approach the data is partitioned into N disjoint subsets, trained on N-1 subsets and the remaining one set is used for validation. This process is repeated for N times and the average accuracy is reported. Cross validation is generally used when the size of the data set is small which is not the case in this research.

The first classifier used is the decision tree. The target attribute is the name of the species to which the sequence belongs. The target attribute in this case is discrete valued, and hence decision trees can be used. The algorithm used is J48 which is an extension to the C4.5 algorithm, developed by Quinlan in 1993. The construction of the tree follows a top down approach. All the attributes are evaluated by a statistical measure called information gain [22]. The attribute with the highest value of information gain will be the root of the tree. The algorithm is then called recursively for every sub-tree. To avoid overfitting of the data the tree is pruned. Overfitting is a phenomenon in which the model is able to predict correct outputs for the training examples but does not generalize well to unseen examples. This usually happens when training is performed for a long time or when the training examples are very rare. It fits the model very well to the training data and hence the performance on the training set increases considerably but it undermines the performance on new unseen data. Similar to decision trees, decision tables also involve selecting the attributes by discarding the irrelevant attributes from the decision table.

The second classifier we used to evaluate the data is Naïve Bayes. It is a supervised learning algorithm in which the target function can take in any discrete value. It uses a probabilistic model where the probabilities of the attributes are calculated. When given a new example to classify the classifier assigns the most probable target attribute value given the other (non-target) attributes [22]. Bayesian network is a probabilistic graphical model that represents a set of random variables and their conditional independencies via a directed acyclic graph. Bayesian networks are similar to naïve Bayes except for the fact that they make weaker assumptions for conditional independance between attributes. Both Naive Bayes and Bayesian networks are relatively fast and take just few minutes to train and classify the instances.

The third classifier used is the multilayered perceptron Artificial Neural Network (ANN) trained by the back-propagation algorithm. ANN's are useful when the training data has lot of noise. The squared error between the target output and the network values is minimized by adjusting the weights [22]. It takes 10 times as long to train a neural network (on the order of few hours on a desktop) as compared to Naïve Bayes which in turn is slower than decision trees. To prevent overfitting of data we changed the default values of the parameters. We used a validation set size of 25%, validation threshold of 20 epochs and trained the network for 500 epochs. This will allow the network to stop on the validation error and not because the number of epochs have been exhausted. We chose these parameters because the domain exhibits a 'saw tooth' behavior. The threshold needs to be kept high and the number of epochs needs to be increased. It may help in such a domain to increase both the learning rate and the momentum term to speed up the training process.

Support Vector Machines (SVMs) are really popular in application domains where the data is linearly separable. It can also be applied to problems of either regression or classification. We used the Sequential Minimal Optimization algorithm which has been implemented in Weka. Training times are faster as compared to ANN but is slow compared to decision trees and Naïve Bayes.

Ensemble methods aim at enhancing the performance of a given statistical model. Ensemble learning is to combine models of either the same type or of a different type. 'Bagging', 'boosting' and 'stacking' all fall into the category of ensemble learners or meta-learners. Bagging and boosting combine models of the same type (usually called the base-classifier type) while stacking combines models of different types (i.e. generated by different algorithms). Bagging is also known as boot strap aggregation. For bagging and boosting we used J48 as the base-classifier because it gave the best performance in the first set of experiments. For stacking we combined a total of five classifiers. The base-classifiers (level-0) used for stacking are Naïve Bayes, SVM, Bayesian network, decision table, and the level-1 or meta-level classifier was J48. There is no hard and fast rule to select which classifiers are to be used for level-0, level-1 classifiers so we selected the faster classifiers because stacking requires much more time than single-classifier learning.

2.3 EXPERIMENTS & RESULTS

2.3.1 Evaluation approach

The performance of our classifiers is tested using validation. We used an 80-20 split of the data, where 80% of the data is used for training/validation and 20% of the data as the test set [23]. The classifier is evaluated on how well it classifies the 20% of the examples based on the learned concept. Other metrics based on which a classifier can be evaluated include Sensitivity, Specificity, and Accuracy. True positives (TP) are the number of positive examples classified as positive. False negatives (FN) are the number of positive examples which are classified as negatives. True negatives (TN) are the number of negative examples which are classified as negative. False positives (FP) are the number of negative examples which are classified as positive [22].

We can define sensitivity and specificity as statistical measures of performance of the binary classification test, namely:

$$\text{Sensitivity} = TP / (TP + FN) \qquad (1)$$

$$\text{Specificity} = TN / (TN + FP) \qquad (2)$$

$$\text{Accuracy} = TP + TN / (TP + FN + TN + FP) \qquad (3)$$

We evaluate our results based on the metric of accuracy, which can be thought as the number of instances which are correctly classified.

2.3.2 Results

Experiments are performed for 3 groups of 15 and 25 species respectively. The number of instances given to the classifiers is approximately 45,000 in the case of 15 species and approximately 77,000 for 25 species. Among meta-learners bagging and boosting with base-classifier as J48 perform better than stacking. Also, training time of J48 is fast as compared to all the other learning algorithms, the worst being ANN which takes hours to build the model.

A closer look at the resulting decision tree reveals that the attribute GC content is always selected as the root of the tree. As GC content is an important feature in determining the functional roles of the species, it has the highest information gain and hence is selected as the root of the decision tree.

Table 2.1 Accuracy of supervised learners on three independent groups of 15 species.

Algorithm	1st Group	2nd Group	3rd Group	Average
J48	92.7218	90.6375	92.3518	**91.9037**
Bayes Net	91.2564	91.0585	92.2176	**91.5108**
Decision Table	92.0281	89.3444	91.0466	**90.8036**
ANN	87.4951	83.1997	85.4111	**85.3686**
NB	81.8875	84.0517	86.4967	**84.1453**
SVM	82.6886	74.4611	83.6545	**80.2680**

Table 2.2 Accuracy of meta-learners on three independent groups of 15 species.

Method	1^{st} Group	2^{nd} Group	3^{rd} Group	Average
Bagging	93.0148	91.4395	93.0471	**92.5004**
Boosting	92.8292	91.5297	93.1447	**92.5012**
Stacking	92.1551	89.9559	91.4979	**91.2029**

As we can see from Table 2.1, decision trees gave the best performance for 15 species. For 25 species, Table 2.3 shows that the Bayesian network performs best but the performance of decision trees is still comparable. Also, when we compared the performance for 15 and 25 species, we expected the accuracy with 15 species to be much better compared with 25 species. This is because in a classification problem with 15 species the random guessing accuracy is expected to be 1/15 or 6.66%, which is higher than that for 25 species (only 4%). However, we noticed that the accuracy for 25 species was not much worse than that of 15 species.

Table 2.3 Accuracy of supervised learners on three independent groups of 25 species.

Algorithm	1^{st} Group	2^{nd} Group	3^{rd} Group	Average
Bayes Net	91.1732	86.0028	86.4445	**87.8735**
Decision Table	91.0554	84.5153	85.7273	**87.0993**
J48	88.2888	86.1439	86.4586	**86.9637**
ANN	84.9694	76.8402	81.9026	**81.2374**
NB	78.5845	75.2437	77.2411	**77.0231**
SVM	76.489	63.5804	69.8024	**69.9572**

Table 2.4 Accuracy of meta-learners on three independent groups of 25 species.

Method	1st Group	2nd Group	3rd Group	Average
Bagging	92.4611	87.2852	87.7522	**89.1661**
Boosting	92.3747	87.5417	87.0491	**88.9885**
Stacking	90.6	84.791	85.6641	**87.0183**

For meta-learners, bagging and boosting perform better than stacking. From the results we can see that the performance of bagging and boosting is higher than J48. We also tried bagging and boosting other classifiers such at the Bayesian network but the best results were obtained with J48 as the base classifier. Also, we can see that the performance of stacking is almost equal to that of J48. From Tables 2.2 and 2.4, we can see that the accuracy of the meta-learners for 15 species is higher than that of 25 species.

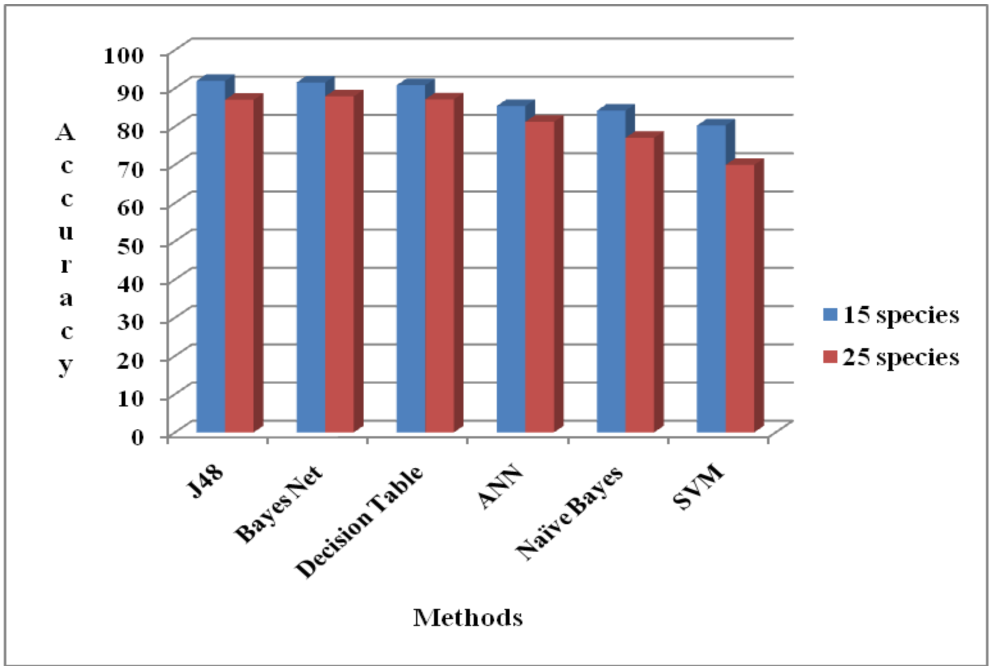

Figure 2.3 Comparison of accuracies of 15 and 25 species for supervised learners.

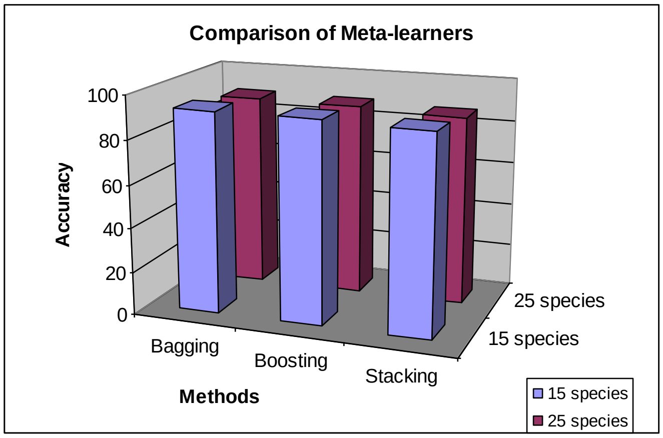

Figure 2.4 Comparison of accuracies of 15 and 25 species for meta-learners.

In [11], the authors used the Naïve Bayes classifier for assigning reads to phylogenetic groups. The features used are k-mer frequencies, and they vary ‘k’ from 7 to 10. They introduced prior knowledge of genomes by a preliminary 16S survey. The Naïve Bayes classifier without any prior knowledge performs with an accuracy of 32.9% for 7-mers and it gradually increases to 43.9% as number of k increases from 7 to 10. To improve the performance they used prior knowledge and the assignment accuracies range from 57.4% to 60.5% as the number of k-mers increase from 7 to 10. For 15 species we have an accuracy of 83.35% using Naïve Bayes, with only five features and without the use of any prior knowledge.

2.4 FUTURE WORK

An extension to the work would be to implement a two phase approach of classifying the sequence reads into respective species first and then by using some statistical analysis determine whether that sequence contains a gene or not. We also plan to evaluate our results by adding more features. Feature selection approaches such as wrapper methods can then be used to improve the performance of the various machine learning algorithms.

Finally, we are currently working on applying our approach to sequence reads which are unknown. We take all the sequence reads present in 15 species and to that set of sequence reads we add reads of sequences from species which are selected randomly and which are distinct from the 15 species that are selected above. We check to see how many of these unknown sequence reads actually get classified as unknown. This way we can simulate a real world problem of mixed sequence reads. We can then evaluate the scalability of the methods and see how well the algorithms perform on such mixed data.

2.5 CONCLUSION

The motivation behind this work is to see which machine learning algorithm does well on classifying or binning the metagenomic data. We address an important problem in the field of bioinformatics, which is to observe relationships among species. Our approach is to apply well known supervised learners and also meta-learners to determine which of the evaluated learning approaches is most accurate on this task. The work is significant as we always have samples of sequences which are mixed or which belong to different species with metagenomic data. When we want to separate the sequences so that they belong to different phylogenetic groups, then this work can help significantly to decide which machine learning algorithms to use.

In this article, we presented a novel approach for classifying the sequences into the respective species by well known machine learning algorithms. The features selected, though only few, were clearly very good at differentiating the data. The accuracy for all the three groups of each number of species (15 or 25) is close, indicating the consistent performance of the learning algorithms. Our results suggest that decision trees can be used to bin the sequence reads in a metagenomic sample. Furthermore, the performance of decision trees can be slightly improved by bagging and boosting. Also, the results indicate that machine learning algorithms such as decision trees, Bayesian networks, decision tables, artificial neural networks, support vector machines, Naïve Bayes can greatly aid in the important task of binning metagenomic data.

2.6 REFERENCES

[1] J. Venter, *et al.*, "Environmental genome shotgun sequencing of the Sargasso Sea," *Science,* vol. 304, p. 66, 2004.

[2] J. Wooley, *et al.*, "A Primer on Metagenomics," *PLoS Computational Biology,* vol. 6, 2010.

[3] T. Dietterich, "Ensemble methods in machine learning," *Multiple classifier systems,* pp. 1-15, 2000.

[4] J. Handelsman, "Metagenomics: application of genomics to uncultured microorganisms," *Microbiology and Molecular Biology Reviews,* vol. 68, p. 669, 2004.

[5] V. Markowitz, *et al.*, "An experimental metagenome data management and analysis system," *Bioinformatics,* vol. 22, p. e359, 2006.

[6] S. Batzoglou, *et al.*, "ARACHNE: a whole-genome shotgun assembler," *Genome Research*, vol. 12, p. 177, 2002.

[7] X. Huang, *et al.*, "PCAP: a whole-genome assembly program," *Genome Research*, vol. 13, p. 2164, 2003.

[8] P. Havlak, *et al.*, "The Atlas genome assembly system," *Genome Research*, vol. 14, p. 721, 2004.

[9] A. Delcher, *et al.*, "Improved microbial gene identification with GLIMMER," *Nucleic Acids Research*, vol. 27, p. 4636, 1999.

[10] G. Tyson, *et al.*, "Community structure and metabolism through reconstruction of microbial genomes from the environment," *Nature*, vol. 428, pp. 37-43, 2004.

[11] L. Kuan-Liang, Tsu-Tsung, Wong, Gary Xie, Nicholas W H, "Improving Naïve Bayesian Classifier for Metagenomics reads assignment," *Biocomp*, pp. 259-264, 2009.

[12] S. McGinnis and T. Madden, "BLAST: at the core of a powerful and diverse set of sequence analysis tools," *Nucleic Acids Research*, vol. 32, p. W20, 2004.

[13] D. Huson, *et al.*, "MEGAN analysis of metagenomic data," *Genome Research*, vol. 17, p. 377, 2007.

[14] J. Wooley and Y. Ye, "Metagenomics: Facts and Artifacts, and Computational Challenges," *Journal of Computer Science and Technology*, vol. 25, pp. 71-81, 2009.

[15] K. Hoff, *et al.*, "Orphelia: predicting genes in metagenomic sequencing reads," *Nucleic Acids Research, vol. 37,* 2009.

[16] H. Noguchi, *et al.*, "MetaGene: prokaryotic gene finding from environmental genome shotgun sequences," *Nucleic Acids Research, vol. 34(19), pp. 5623 - 5630,* 2006.

[17] H. Noguchi, *et al.*, "MetaGeneAnnotator: detecting species-specific patterns of ribosomal binding site for precise gene prediction in anonymous prokaryotic and phage genomes," *DNA research,* vol. 15, p. 387, 2008.

[18] A. Lukashin and M. Borodovsky, "GeneMark. hmm: new solutions for gene finding," *Nucleic Acids Research*, vol. 26, p. 1107, 1998.

[19] *Comprehensive Microbial Resource*. [Online]. Available: http://cmr.jcvi.org/cgi-bin/CMR/shared/Menu.cgi?menu=downloads. [Accessed: April 11, 2010].

[20] J. Raes, *et al.*, "Get the most out of your metagenome: computational analysis of environmental sequence data," *Current opinion in microbiology,* vol. 10, pp. 490-498, 2007.

[21] *WEKA 3- Data Mining with open source Machine Learning software in Java.* [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/. [Accessed: April 11, 2010].

[22] T. M. Mitchell, *Machine Learning*: McGraw-Hill, 1997.

[23] P. Crowther and R. Cox, "A Method for Optimal Division of Data Sets for Use in Neural Networks," *Springer Berlin/Heidelberg,* pp. 1-7, 2005.

CHAPTER 3

ON THE SCALABILITY OF SUPERVISED LEARNERS IN METAGENOMICS[2]

[2] Vasim Mahamuda, Manchon U, Khaled Rasheed, Submitted to The Ninth International Conference on Machine Learning and Applications (ICMLA 2010), 7/13/2010.

3.0 ABSTRACT

Metagenomics deals with the study of micro-organisms such as prokaryotes that are found in samples from natural environments. The samples obtained from the environment can include DNA from many different species. Metagenomic data can include bacteria and archea. Micro-organisms are responsible for most of the symbiotic activity on earth. They are also responsible for the complex chemical reactions which take place on the surface of the earth, which help maintain earth's ecological balance. With the increase in genome sequencing projects there has been a considerable increase in the amount of assembled sequencing data. In this article, we apply supervised learners namely decision trees, Bayesian networks and decision tables to see how the performance degrades when the number of species present in the metagenomic sample increases. We also try to see how the performance of the metagenomic sample changes as the percentage of unknown sequences in the metagenomic sample is varied.

Keywords: Bayesian networks, binning, bioinformatics, decision trees, machine learning, metagenomics.

3.1 INTRODUCTION

Prokaryotic microbes, including Bacteria and Archea, are found in all diverse environments on Earth, ranging from soil, seawater or human intestine to deep-water hydrothermal vents characterized by extreme pressure and temperature. They are essential to all life on our planet. Metabolic activities of prokaryotes helped shape the Earth's environment to be able to support higher forms of life and many eukaryotic organisms rely on prokaryotic symbionts for survival. Until recently, the DNA sequencing of prokaryotic genomes was limited to those that could be cultivated in the laboratory. However, many prokaryotes cannot be cultivated outside their natural environment, which often involves complex microbial communities.

In order to facilitate the study of uncultivated micro-organisms a new field known as 'metagenomics' has emerged in the area of genetic research [1]. Metagenomics allows us to study microbial communities in order to understand the roles they play in the environment, in our own bodies, or as symbionts of plants and animals.

Metagenomic data is obtained from DNA samples extracted from various environments, such as sea water, land and human guts. The DNA is sheared into small fragments, which are randomly "sequenced". That is, the exact sequence of nucleotides in the fragment is determined. Those nucleotide sequences are often referred to as 'fragments' or 'reads'. Some of the most popular sequencing methods are Sanger sequencing and 454 sequencing. Nowadays, large scale sequencing projects yield sequences which are between 300-1000 nucleotides in length.

Analysis of the nucleotide sequences generated by metagenome sequencing projects represents one of the major computational challenges of the present bioinformatics. The data typically consists of hundreds of thousands of individual sequence reads, which originated from many different organisms present in the original sample. At the same time, the amount of DNA sequenced often represents only a small fraction of all DNA in the sample. To make the task more complicated, the sequence reads can contain a small number of random errors. The main tasks in the analysis of metagenomic data involve assembly of overlapping reads into larger contigs, identification of genes present in the DNA sample, and the phylogenetic classification of the sequence reads or contigs.

This work centers on the phylogenetic classification of metagenomic sequences. The task aims to assign each sequence read or contig to the species from which the DNA originated. There are principally two techniques applicable to this task: (a) sequence-similarity based and (b) sequence-composition based classification. Sequence-similarity methods such as BLAST [2] and MEGAN [3] use sequence alignment to assign the assembled contigs to reference genomes. If a particular read closely resembles a sequence in the reference database, one can assume that this read originated from the same or related species to that of the matching sequence in the database. Sequence-composition based techniques such as clustering methods, group the sequences based on oligonucleotide composition of the assembled contigs. Clustering methods such as K-means can be used to cluster the contigs into groups or bins. There is no necessity for any kind of reference genomes to assign these contigs to bins in unsupervised learning [4]. The term ‘bin’ here refers to assigning contigs to phylogenetic groups and the process is termed as ‘binning’. Sequence similarity based techniques can be categorized as supervised learners while sequence composition based techniques can be classified as unsupervised learners.

The significance of mapping contigs to the phylogenetic tree is to understand the functional and biological roles of these molecules in the environment. Mapping contigs to taxonomic classification helps to know the composition of these species in the environment, and also we can predict the roles of these assembled genes by looking at the community to which the reads belong [5]. As discussed earlier, this kind of mapping can be achieved either through supervised or unsupervised learning.

With the invention of new sequencing technologies there has been a rapid increase in the number of sequences available in the biological databases. Metagenomics is a real world problem, where the number of sequences (data set) increases and there are only few of them which actually have reference genomes. So, there is always a need for some probabilistic model, which can infer the genus or phyla that the newly sequenced reads belong to. Using the probabilistic models we can also infer whether a particular sequence includes an encoded gene. Also, the need for computational methods arises while dealing with such humungous amounts of data. Machine learning helps in fields like metagenomics because in the real world there is a huge amount of data [6]. Machine learning is capable of learning from any kind of labeled or unlabeled data.

Machine learning algorithms can also be used for predicting novel genes. There are many gene finding programs that are available for predicting genes in prokaryotic sequences. One of them is Orphelia [7] which is available as a web server. The Orphelia program uses linear discriminants to decrease the number of features and then uses artificial neural networks to predict genes in the metagenomic sequence reads. Another gene finding program for microbial genomes is the GeneMarkS [8] which uses a hidden markov based model. In [8], the authors claim that their method can be used for finding genes in prokaryotic genomes without prior knowledge of any proteins. In [4], the authors used a naïve Bayesian classifier to bin the metagenomic sequences into the respective phylogenetic groups. The other prokaryotic gene finding algorithm is MetaGene [9] which uses two different methods of feature extraction for bacteria and archea. The algorithm is programmed such that it switches between the two methods according to the given input sequence. The sequences used in the method are taken from the Sargasso Sea data set. In [9], the authors were able to predict novel genes in addition to the annotated genes. In [10], the authors use an incremental clustering approach where the data is passed through various stages of clustering. They were able to predict the novel genes in metagenomic sequences and also group sequences into various families.

The work in this article is an extension to our previous work in this field [11]. In [11], we applied several supervised learners and meta-learners to sets of 15 and 25 species to see which machine learning algorithms perform well on such metagenomic data. A total of six supervised learners were used namely - decision trees, Naïve Bayes, support vector machines, artificial neural networks, Bayesian networks and decision tables. The three meta-learners used were bagging, boosting and stacking. Results from [11], suggest that decision trees, Bayesian networks and decision tables perform better than the other learners, and they were also able to classify sequences to their respective species with a high percentage of accuracy. For a set of 15 species, 91.9% of the examples were correctly classified and for a set of 25 species 86.9% of the examples were correctly classified with the decision tree classifier. In this research, we go further to see how the performances of the classifiers vary with regard to scalability.

The main goal of this article is to investigate the application of supervised learning models namely - decision trees, Bayesian networks, decision tables to metagenomic samples which have different number of species to see how the performance degrades as the number of species in the sample increases from 15 to 300 and also to see how the performance of the classifier changes as the number of unknown sequences in the sample increases. We achieve this by taking a set of metagenomic sequences and attempting to 'bin' them. In simple terms, we are trying to classify the sequences into their respective 'groups'. Groups here refer to species into which these sequences are to be mapped. Here, we are trying to model a problem of classification where the sequences need to be classified or mapped to a set of discrete values. We extract five features from the metagenomic sequences, such that they are representative of the underlying data.

The rest of the article is organized as follows: In Section 3.2 we introduce our methodology and show how the features are extracted from the sequence data. In Section 3.3 we present the performance of the classifiers. In Sections 3.4 and 3.5 we discuss future work and conclusion.

3.2 EXPERIMENTAL METHODOLOGY

In this section, we briefly describe the classifiers used in our experiments and the algorithm used to extract features from the raw assembled sequence data.

3.2.1 Learning Methods

Machine learning is a branch of artificial intelligence. Machine learning algorithms can be applied to a myriad of problems ranging from board game playing, face recognition, prediction of diseases … etc. These algorithms have been successfully applied to real world problem from a variety of fields. The term ‘learning’ can be elucidated as gathering some new knowledge and improving it by connecting the gathered information to what has been previously known. One form of learning which machine learning deals with is to gather knowledge in the form of some abstract concepts and then use those concepts to build an overall knowledge base of the given problem [12]. In simple terms if a problem has a set of examples which represent the underlying concept, the machine learning algorithm learns the concept based on these examples and when challenged with a query it predicts the outcome of the query. The way these algorithms learn a concept based on some past experience or knowledge is really exceptional. Machine learning algorithms can be applied either to the problem of classification or regression. In classification we have a set of discrete outcomes whereas, regression involves mapping the inputs to values in a specified continuous range. The outcome is also known as ‘target attribute’, the one to which the input examples must be mapped.

The set of examples presented to the machine learning algorithm is known as the ‘training set’. Examples in this set consist of a set of attributes or features which represent the underlying concept. One of the attributes in the list of attributes is known as the ‘target attribute’ also known as the label or outcome. The query examples are given to the algorithm in form of a ‘testing set’. The ratio of training and testing sets greatly influences the performance of the algorithm. The performance drops if very few examples are presented to learn the concept. To prevent over-fitting, another set called a validation set is often used. The training set is randomly split into training and validation sets and the algorithm is trained on one set and is tested on the other. According to [13], the optimal split for data is to have 1/3rd data for validation and the remaining ratio of train/test set can range anywhere between 50/50 to 70/30. The classifiers which we use in our experiments are decision trees, Bayesian networks and decision tables.

Decision trees are capable of classifying discrete valued target attributes. The algorithm used is J48 which is an extension to the C4.5 algorithm, developed by Quinlan in 1993. The construction of the tree follows a top down approach, at each node it chooses the attribute which has a high score which is evaluated by a heuristic known as information gain. The attribute with the highest value of information gain, among all the attributes will be the root of the tree [14].

Let us define the term information gain in more detail. Before we define information gain we first define a measure called ‘entropy’. Given a set of examples N whose target attribute has two outcomes (‘yes’ or ‘no’). Entropy of N relative to this binary classification is defined as:

$$\textit{Entropy (N)} \equiv - y \log_2 y - n \log_2 n \qquad (1)$$

where, y is proportion of examples labeled as ‘yes’ and n is the proportion of examples labeled as ‘no’.

If the target attribute takes more than two values i.e multi class classification then the entropy of N relative to multi class classification of ‘c’ number of classes is defined as:

$$\textit{Entropy (N)} \equiv \sum_{i=1}^{c} -y_i \log_2 y_i \qquad (2)$$

where, y_i is the proportion of N belonging to class i, where ‘i’ ranges from 1 to c.

Information gain *Gain(N,F)* of a feature F, relative to collection of examples N is defined as:

$$Gain(N,F) \equiv Entropy\ (N) - \sum (|\ N_v\ |\ /\ |\ N|)\ Entropy\ (N_v) \qquad (3)$$

where, v belongs to *Values(F*) is the set of all possible values for feature F, and N_v is the subset of N for which the feature F has value v.

Information gain is calculated for all the attributes. The attribute with the highest information gain is selected as the root of the tree. This procedure is repeated recursively and information gain is the measure which decides which attribute should be the root of the corresponding sub-tree. Similar to decision trees, decision tables also model complex logic through simple conditions similar to if-then-else statements. Bayesian networks deal with set of random variables and their conditional dependencies which are represented through a probabilistic graph. All the three above mentioned classifiers are fast and usually just take a few minutes to train and classify thousands of instances.

3.2.2 Data Set

The data used in the experiments was downloaded from the Comprehensive Microbial Resource [15]. The assembled contigs downloaded from the website need not go through biological processing such as genome sequencing or gene assembly. The data is already processed and the sequence reads are in the form of a FASTA file for each individual species. A FASTA file has many fragments of sequences from the same species. The sequence reads in a FASTA file usually start with a sequence identifier followed by the sequence itself. The number of fragments of sequence reads present in a single FASTA file depends on the species selected. The average number of reads of DNA sequences in a file of each species would be approximately 3500 nucleotides and the length of DNA sequence would be between 300-1000 nucleotides. In this work we create synthetic metagenomic data by mixing a large number of fragments from different species as we do not have access to real metagenomic data.

Two sets of experiments are performed. The first set of experiments deals with varying the number of species in the sample. The second set of experiments is performed by varying the percentage of unknown sequences in the sample. The data for the first set of experiments are selected such that there are a total of 300 species. The data for second set of experiments are selected in such a way that we have three different sets called the known set, unknown set (train), unknown set (test). For the known set we selected 25 species. For the unknown set (train) we selected 25 species. For the unknown set (test) we selected 50 species. The species were randomly selected for the three sets and are distinct from each other.

3.2.3 Feature Extraction

The features we used are the number of ORFs, uni-base frequency, di-base frequency, GC content, average length of ORF. We find the 'Open Reading Frames' or ORFs for each sequence read. If a start codon and a stop codon are separated by at least 54 base pairs, then we consider it as an ORF [7]. Codons are triplets in biology. If a codon has a pattern such as ATG, CTG, GTG or TTG it is called a start codon. Similarly if it has the pattern TGA, TAG or TAA it is called a stop codon. As a sequence can be translated in six different ways, we find the ORFs in all possible frames (3 on the positive strand, 3 on the minus strand). In each ORF we find what we call the "uni-base frequency" and the "di-base frequency". Uni-base frequency is the term we use for codons which contain one unique nucleotide, namely AAA, TTT and so on. Di-base frequency is the term we use for codons which contain only two unique nucleotide, namely CCG,GGC and so on [11]. The algorithm which we use to process the data is simple and works as follows. For each sequence read we calculate the number of ORFs, uni-base frequency, di-base frequency, GC content and average length of ORFs.

We generate the features using the above algorithm and then use Weka [16] to run the classifiers. Weka is a suite of java packages which has implementations of various machine learning algorithms. A nice feature of Weka is that we can call all the classes from within our java code.

3.3 RESULTS

In this section, we describe how we evaluate the performance of the classifiers. We chose decision trees, Bayes Net and decision tables as the learning methods. We have two sets of experiments. We first describe the evaluation for the first set of experiments and then for the second set.

For the first set of experiments, we used an 80-20 split of the data, where 80% of the data is used for training/validation and 20% of the data as the testing set. The classifier is evaluated on how well it classifies the 20% of the instances in the testing set. Metrics based on which a classifier can be evaluated include Sensitivity, Specificity, and Accuracy. We calculate the accuracy as the measure of performance for our experiments. The accuracy of a classifier is the percentage of instances for which the classifier predicts the correct target attribute value in the testing set. The main goal of these experiments is to see how well the metagenomic sequences get mapped to their respective species from which the DNA samples originated.

Table 3.1 Accuracies of the classifiers for different values of species.

Number of Species	J48	Bayes Net	Decision Table
15	88.61	88.58	89.83
25	85.17	85.17	84.45
35	77.25	78.04	77.11
45	75.65	76.44	75.13
55	73.61	74.6	73.61
65	71.52	72.84	71.76
75	67.85	69.35	68.19
100	55.43	58.36	57.24
150	47.69	52.17	51.08
200	41.79	46.66	45.41
300		42.38	40.8

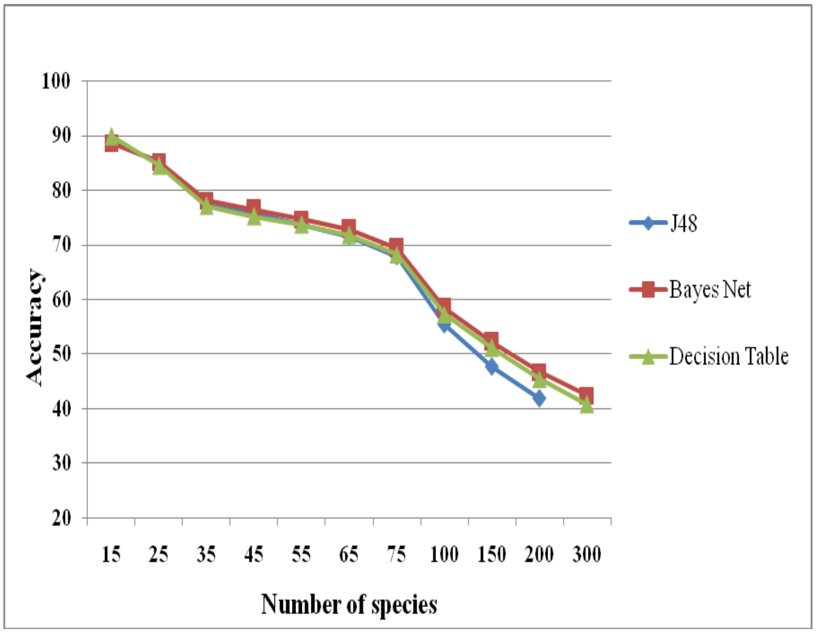

Figure 3.1 Performance variation with the increase in the number of species.

The experiments are carried out in the following way: We varied the number of species in the sample from 15 to 300. The number of instances varied from 30,000 for 15 species to an order of 800,000 instances for the set of 300 species. The training/testing times also varied according to the number of instances. Among the three classifiers, decision trees are a bit slower compared to other learning methods.

As we can see from Figure 3.1, the performance of all the three classifiers is almost the same for any value of species, indicating the consistent performance of all three classifiers. Also, we notice that the performance of the classifiers drops as the number of species increases from 15 to 300. Table 3.1 shows the accuracy of each classifier for different number of species. From Figure 3.1, we notice that the performance of J48 begins to degrade as compared to other algorithms as the number of species in the sample increases from 75 to 200. We do not have the accuracy value of J48 for 300 species as it takes too long to run on a standard desktop machine.

We now describe the second set of experiments. We have three sets of species namely - known set, unknown set (train) and unknown set (test). We have two experiments here, one with 15 species in the known set and another with 25 species in the known set. For the 15 species in the known set, the training file consists of 95% sequences from the known set and 5% sequences from the unknown set (train). We label the 95% sequences which are taken from the known set with their respective species names and the 5% sequences taken from the unknown set (train) are labeled as ‘unknowns’. For the testing file we take sequences from the known set and unknown set (test), such that we have a series of 8 different test files where the percentage of sequences from the unknown set (test) vary from 15% to 90%. For each of the testing files the species in the known set are labeled with their respective species names and sequences from the unknown set (test) are labeled as ‘unknowns’. In this way, we have a total of 16 discrete classes (15 labels for 15 species in the known set + one label ‘unknown’ for species in the unknown set) into which these instances are to be classified. The classifiers are trained with the training file and are then tested on the 8 different test files.

Table 3.2 Accuracy of the classifiers for set of 15 species in the known set.

% of unknown sequences	J48	Bayes Net	Decision Table
15	60.75	65.17	62.79
25	58.47	61.65	59.28
40	56.57	58.42	56.01
50	54.84	56.41	53.73
60	53.84	54.96	52.14
70	52.99	54.01	50.92
80	52.43	52.79	49.34
90	51.31	51.17	47.15

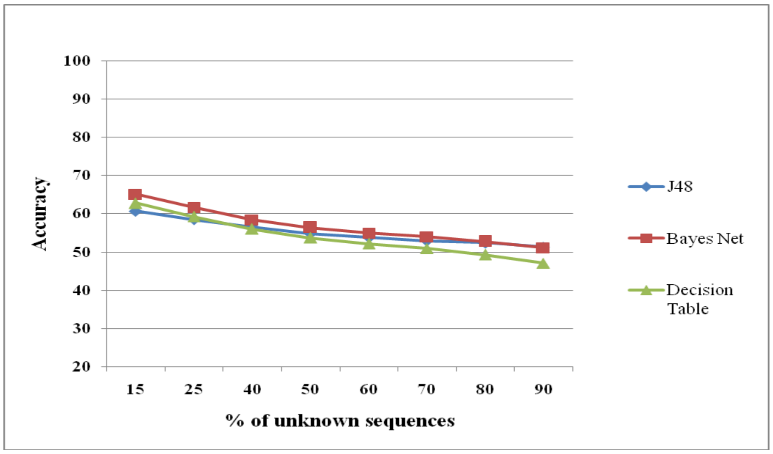

Figure 3.2 Performance variation with the increase in the number of unknown sequences with 15 species in the known set.

The same experiment is repeated with 25 species in the known set. We kept the experiments to the sequence level. For the experiment with 15 species in the known set, we fixed the number of sequences to 30,000 and for the experiment with 25 species the number of sequences is 50,000. From Figures 3.2 & 3.3, we can see that the accuracy of the classifier decreases as the percentage of unknown sequences in the sample increases. Also, we can observe that when the sample has no unknown sequences the accuracy of the classifier is 93% with 15 species and 81% with 25 species indicating the high performance of the classifiers even though the features used were only five. Tables 3.2 & 3.3, show the accuracy of classifiers for set of 15 and 25 species in the known set.

Table 3.3 Accuracy of the classifiers for set of 25 species in the known set.

% of unknown sequences	J48	Bayes Net	Decision Table
15	58.31	59.46	58.30
25	56.97	57.99	56.82
40	55.89	56.91	55.88
50	55.36	56.56	55.54
60	53.61	54.98	54.2
70	52.5	54.35	53.8
80	50.99	53.04	52.42
90	49.38	51.39	50.73

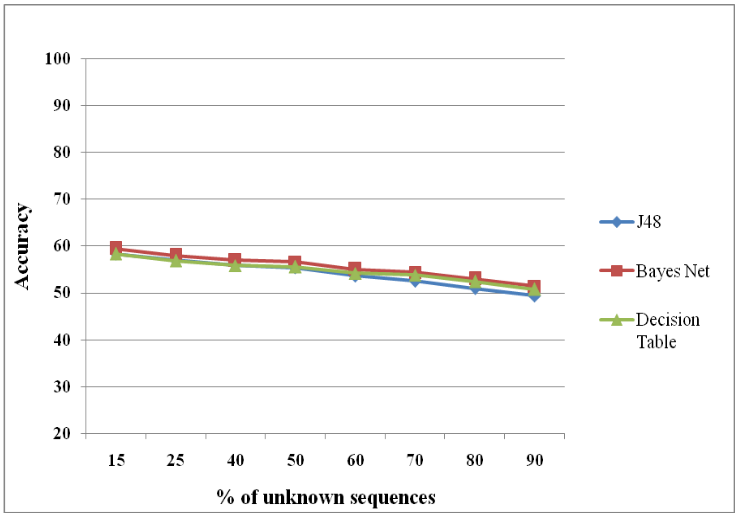

Figure 3.3 Performance variation with the increase in the number of unknown sequences with 25 species in the known set.

3.4 FUTURE WORK

An extension to the work would be to add more attributes. New features such as k-mer frequencies and octo-nucleotides can be added which are representative of the underlying data. We can then apply different feature selection methods such as wrapper based or filter based approaches to see how the classifiers perform. Wrapper based methods search through an entire set of features and evaluates each subset by running a model for each subset. Filter based approaches apply a simple filter to form a subset of features rather than evaluating a classifier on those features.

For the scalability study with regard to species we considered only one level of classification, i.e classifying the sequence reads into species. We can extend the binning process to higher levels in the taxonomy such as the family or the class to which a sequence belongs. As we go to a higher level of taxonomy the probability that the sequence will be classified into correct taxonomic group is higher. An explanation to this is that sequences which belong to the same group or taxa, have more probability of getting grouped together. Also, we can try to apply the meta learners namely bagging, boosting and stacking to see how their performance scales.

3.5 CONCLUSION

The motivation behind this work is twofold; one is to see how the performance of the classifiers degrades when the number of species in the sample increases and another is to see how the performance of the classifiers varies when the number of unknown sequences in the sample changes. The work is significant as we tried to mimic a real world problem of metagenomics, by considering the fact that in a metagenomic sample there is little knowledge or no knowledge of the species present in the sample. They can either be related to each other or might be completely different. We notice that the performance of the classifiers drops when the number of species in the sample increases from 15 to 300. This can be attributed to the fact that when the number of species in the sample increases there is more chance that the species might be related and in turn this decreases the performance of the classifier as it has more confusion in classifying them correctly.

In this paper we presented experiments in the context of classifying the sequences into the respective species with the help of decision trees, Bayesian networks, and decision tables. All the three learners are fast. The selection of the algorithms and features used was good enough as we were able to bin the sequences with a percentage of accuracy much higher than the expected random guessing accuracy. Training and testing of approximately 50,000 instances just took an order of a few minutes. The features selected, though very few, were good at differentiating the data. Finally the results are very promising to the metagenomic researcher as the performance degraded very gracefully with the increase in the number of species as well as the increase in the proportion of unknown sequences.

3.6 ACKNOWLEDGMENT

We would like to thank Dr. Katharina Hoff, for her time and effort. Dr. Hoff provided us with the background knowledge necessary for applying computational skills to metagenomes. We would also like to thank Eric Talevich for his help with some biological details. Also, we would like to thank Dr. Jan Mrazek for his suggestions and insightful thoughts which helped us to improve the article.

3.7 REFERENCES

[1] C. Riesenfeld, *et al.*, "Metagenomics: genomic analysis of microbial communities," vol. 38, pp. 525-552, 2004.

[2] S. McGinnis and T. Madden, "BLAST: at the core of a powerful and diverse set of sequence analysis tools," *Nucleic AcidsResearch*, vol. 32, p. W20, 2004.

[3] D. Huson, *et al.*, "MEGAN analysis of metagenomic data," *Genome Research*, vol. 17, p. 377, 2007.

[4] L. Kuan-Liang, Tsu-Tsung, Wong, Gary Xie, Nicholas W H, "Improving Naïve Bayesian Classifier for Metagenomics reads assignment," *Biocomp*, pp. 259-264, 2009.

[5] J. Raes, *et al.*, "Get the most out of your metagenome: computational analysis of environmental sequence data," *Current opinion in microbiology,* vol. 10, pp. 490-498, 2007.

[6] Z. Ghahramani and M. Jordan, "Supervised learning from incomplete data via an EM approach," *Advances in Neural Information Processing Systems 6,* 1994.

[7] K. Hoff, *et al.*, "Orphelia: predicting genes in metagenomic sequencing reads," *Nucleic Acids Research, vol. 37,* 2009.

[8] J. Besemer, Lomsadze,A. and Borodovsky,M., "GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions," *Nucleic Acids Res*, vol. 29, pp. 2607-2618, 2001.

[9] H. Noguchi, *et al.*, "MetaGene: prokaryotic gene finding from environmental genome shotgun sequences," *Nucleic Acids Research,* vol. 34(19), pp. 5623 - 5630, 2006.

[10] S. Yooseph, W. Li, *et al.,* "Gene identification and protein classification in microbial metagenomic sequence data via incremental clustering," *BMC Bioinformatics,* vol. 9, 2008.

[11] Vasim Mahamuda, Manchon U, Khaled Rasheed, "Application of Machine Learning Algorithms for Binning Metagenomic Data," To appear in Proceedings of The International Conference on Bioinformatics and Computational Biology *BIOCOMP'2010,* 2010.

[12] J. Quinlan, "Induction of decision trees," *Machine learning,* vol. 1, pp. 81-106, 1986.

[13] P. Crowther and R. Cox, "A Method for Optimal Division of Data Sets for Use in Neural Networks," *Springer Berlin/Heidelberg,* pp. 1-7, 2005.

[14] T. M. Mitchell, *Machine Learning*: McGraw-Hill, 1997.

[15] *Comprehensive Microbial Resource*. [Online]. Available: http://cmr.jcvi.org/cgi-bin/CMR/shared/Menu.cgi?menu=downloads. [Accessed: June 15, 2010].

[16] *WEKA 3- Data Mining with open source Machine Learning software in Java.* [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/. [Accessed: June 15, 2010].

CHAPTER 4

CONCLUSION

In this thesis, we present two experimental studies to analyze metagenomic data. In our first set of experiments we tried to see which machine learning algorithm performs well on metagenomic data. To do this we developed a DNA-composition-based algorithm that we have used for processing simulated metagenomic sequence reads. We tried a variety of popular machine learning algorithms representative of the basic machine learning approaches such as decision trees, naïve Bayes, artificial neural networks, support vector machines, Bayesian networks, and decision tables. We also tried ensemble learning methods namely – bagging, boosting and stacking. We conducted the experiments for classifying sequences from sets of fifteen and twenty five species.

Results show that decision trees, Bayesian networks, and decision tables perform much better than other popular algorithms. To validate our approach we used three different data sets and then averaged their accuracy. For a set of 15 species, we have an accuracy of 91.9% and for a set of 25 species 86.9% of the examples were correctly classified with decision trees. From this study we were able to infer that with the features we selected decision trees were able to classify the sequences with very high accuracy.

In the second set of experiments, we evaluated the performance of the classifier with respect to scalability with the number of species. We used the classifiers decision trees, Bayesian networks, and decision tables, the winners of the first study, to see how the performance of the classifier varies as the number of species in the metagenomic sample increase from 15 to 300 species. It is obvious and expected that the performance of the classifier degrades as the number of species in the sample increase. But even with 300 species we have an accuracy of 42.38% with the Bayesian network which is a very promising result for the metagenomic researchers. The random guessing accuracy with 300 species would be only 1/3%. The graph below shows a plot of the ratio between the actual accuracy and the random guessing accuracy for 25, 100, 150, 200 and 300 species. We name this measure as 'relative learning gain' of the classifier.

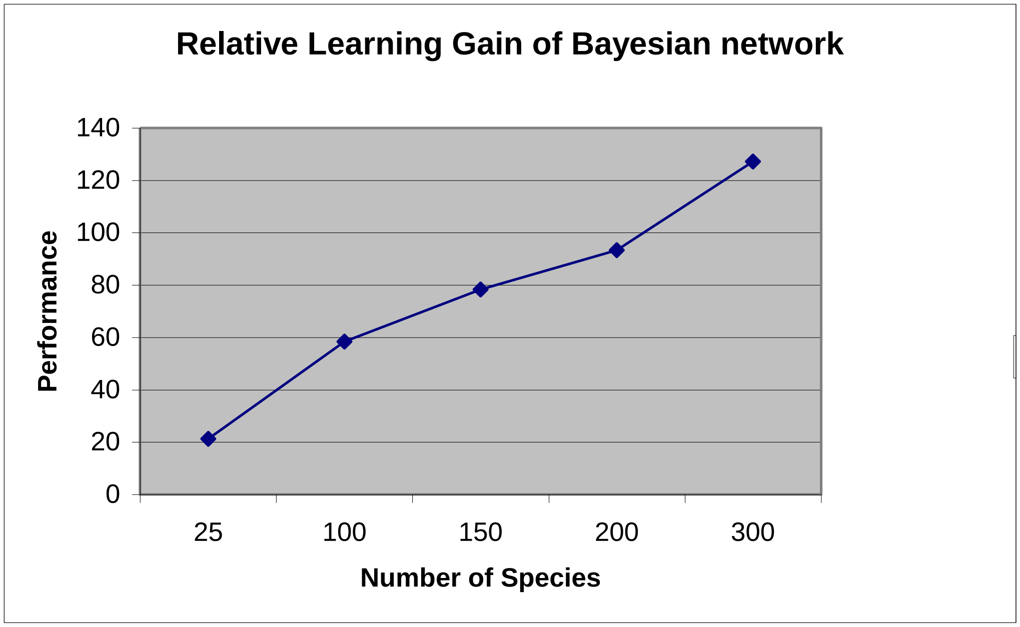

Figure 4.1 The relative learning gain of the Bayesian network classifier for different values of species.

We conducted additional analysis using a set of 75 species. We used a two phase approach where we first classified the sequences into their respective genus and then into species. The set of 75 species was selected such that we have 15 different genera, and each genus has a set of 5 species. By doing a two phase approach we have a series of 16 different classifiers, i.e. one classifier for classifying the sequences into genera and 15 classifiers for further classifying sequences which belong to each genus into species. Figure 4.2, shows the performance for the (direct) single phase approach and the two phase approach. For the two phase approach we calculated the true positive rate for each species and averaged all the 75 values. The true positive rate for each species was calculated as (True positive rate of genus) X (True positive rate of species).

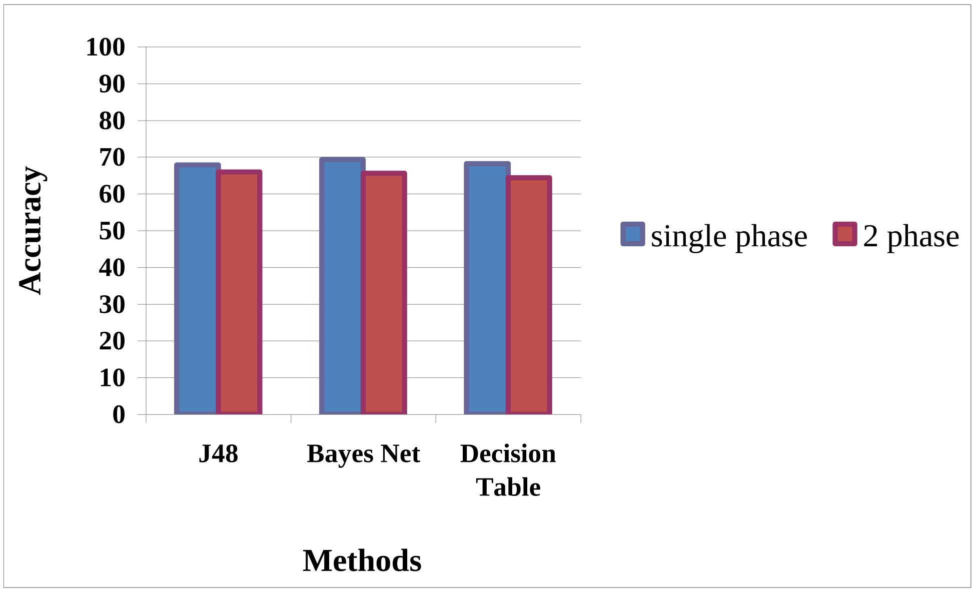

Figure 4.2 Performance of single phase and two phase approach for a set of 75 species.

From figure 4.2., we can see that the two phase approach did worse than the single phase approach. The failure of the two phase approach might be due to the error cascading problem. If the genus classification is incorrect the species classification would certainly be wrong. The only way a correct classification is made is when both the genus and the species classifiers work correctly.

We also tried to simulate a real world problem of mixed sequences by introducing unknown sequences in the test files. The results were very good even though we included a random set of 50 species in the test files.

We did not include any kind of prior knowledge in our experiments. The selection of the data was random but with a limitation that each species selected belonged to a different genus in the first approach. We have not limited ourselves to just one classifier but tried a variety of different classifiers, to check which classifier performs well on metagenomic data. The approach we used is different from the recent work done in this area as we are trying to classify the sequences into a known set of discrete valued species rather than trying to find genes in a particular sequence.

This work helps us to know which machine learning algorithm to use and what features help in classifying the sequences. When we want to separate the sequences such that they belong to different phylogenetic groups then this work can help significantly to decide which machine learning algorithms to use. The separation of the assembled contigs into taxonomic bins helps to know the composition of the environment. With the features used, we were able to assign the sequences to their respective groups with a high percentage of accuracy. Also, the results suggest that researchers in the metagenomic community should consider learning algorithms like decision trees and Bayesian networks to bin metagenomic sequences rather than using popular methods like support vector machines or aritificial neural networks. Also, artificial neural networks take a long time to classify the sequences whereas, with Bayesian networks training and testing of instances just takes in the order of a few seconds. We would like to conclude by saying that several learning methods including decision trees, Bayesian networks and decision tables proved to be useful to the metagenomics researcher and their performance degrades gracefully with the increase in the number of species and the percentage of unknown sequences in the metagenomic sample.

REFERENCES

[1] J. Wooley, *et al.*, "A Primer on Metagenomics," *PLoS Computational Biology, vol. 6,* 2010.

[2] C. Riesenfeld, *et al.*, "Metagenomics: genomic analysis of microbial communities," vol 38, pp. 525-552, 2004.

[3] K. Mavromatis, *et al.*, "Use of simulated data sets to evaluate the fidelity of metagenomic processing methods," *Nature methods,* vol. 4, pp. 495-500, 2007.

[4] S. McGinnis and T. Madden, "BLAST: at the core of a powerful and diverse set of sequence analysis tools," *Nucleic Acids Research,* vol. 32, p. W20, 2004.

[5] D. Huson, *et al.*, "MEGAN analysis of metagenomic data," *Genome Research,* vol. 17, p. 377, 2007.