On the Scalability of Supervised Learners in Metagenomics

ManChon U
Department of Computer Science
The University of Georgia
Athens, GA 30602, USA
manchonu@cs.uga.edu

Vasim Mahamuda
Department of Computer Science
The University of Georgia
Athens, GA 30602, USA
mahamuda@cs.uga.edu

Khaled Rasheed
Department of Computer Science
& Institute for Artificial Intligence
The University of Georgia
Athens, GA 30602, USA
khaled@uga.edu

***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 may contain DNA from many different species of micro-organisms including 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.

I. 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, and 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]. 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].

The rest of the article is organized as follows: We introduce the related work in section II. In section III we illustrate our methodology and show how the features are extracted from the sequence data. In section IV we present the performance of the classifiers. In sections V and VI we discuss future work and conclusion.

II. Related Work

Metagenomics is a real world problem, where the number of sequences increases and there are only a few of them which actually have reference genomes. Therefore, there is always a need for some probabilistic model, which can infer the genus or phyla that the newly sequenced reads belong to. Machine learning performs well in this type of problem domain as it is capable of learning from a large set of labeled or unlabeled data for classification [6]. Utilizing the learned models, we can also infer whether a particular sequence includes an encoded gene. 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. The Orphelia program [7] uses linear discriminants to decrease the number of features and then uses artificial neural networks to predict genes in the metagenomic sequence reads. In GeneMarkS [8], the authors claim that their method (hidden markov model) 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 which 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 were able to classify sequences to their respective species with a high degree 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.

III. 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.

A. 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. 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 problems 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 1/3 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. Given a set of examples S whose target attribute has two outcomes ('yes' or 'no'). The Entropy of N relative to this

binary classification is defined as:

$$Entropy(S) \equiv -p_{\oplus} log_2 p_{\oplus} - p_{\ominus} log_2 p_{\ominus} \quad (1)$$

where $p_{\oplus}$is the proportion of positive examples in S and $p_{\ominus}$ is the proportion of negative examples in S. If the target attribute takes on c different values, then the entropy of S relative to multi class classification is defined as:

$$Entropy(S) \equiv \sum_{i=1}^{c} -p_i log_2 p_i \quad (2)$$

where, p_i is the proportion of S belonging to class i, where 'i' ranges from 1 to c.

Information gain Gain(S,A) of an attribute A, relative to collection of examples S is defined as:

$$Gain(S, A) \equiv Entropy(S) - \sum \frac{|S_v|}{|S|} Entropy(S_v) \quad (3)$$

where, v belongs to Values(A) is the set of all possible values for feature A, and S_v is the subset of S for which the feature A 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 a set of random variables and their conditional dependencies which are represented through a probabilistic graph. All three above mentioned classifiers are fast and usually just take a few minutes to train and classify thousands of instances.

B. Data Set

The data used in the experiments was downloaded from the Comprehensive Microbial Resource [15]. 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.

C. 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.

IV. RESULTS

In this section, we describe how we evaluate the performance of the classifiers and the results of two sets of experiments. We chose decision trees, Bayes Nets and decision tables as the learning methods.

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% for 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. 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 1, the performance of all 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 I shows the accuracy of each classifier for different number of species. From Figure 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.

TABLE I. 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	74.60	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	37.56	42.38	40.80

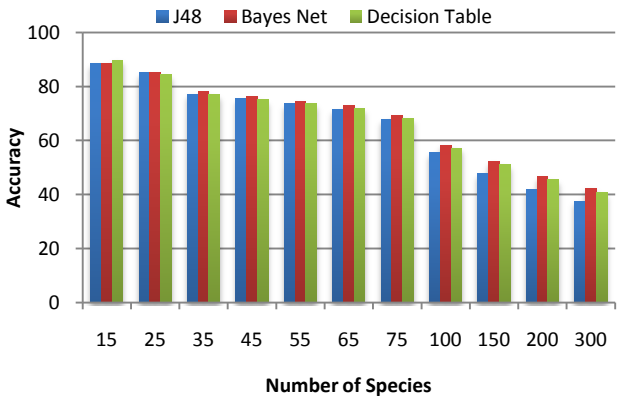

Figure 1. Performance of classifiers with different number of species

In the second set of experiments, we have three sets of species namely - known set, unknown set (train) and unknown set (test). We did two experiments, 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.

The same experiment is repeated with 25 species in the known set. 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 2 & 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 II & III, show the accuracy of classifiers for set of 15 and 25 species in the known set.

TABLE II. ACCURACY OF THE CLASSIFIERS FOR SET OF 15 SPECIES IN THE KNOWN SET WITH VARIOUS % OF UNKNOWN SEQUENCES

% 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

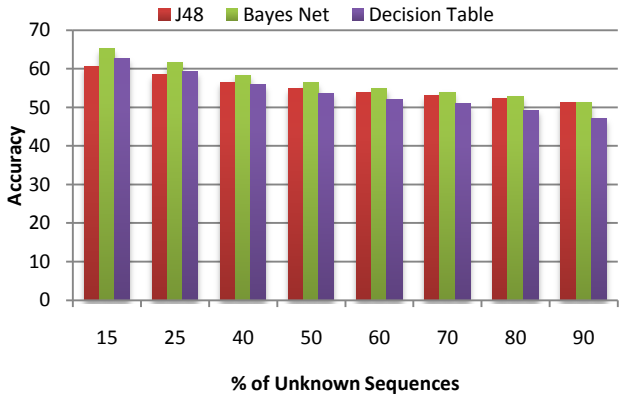

Figure 2. Performance of classifiers with different % of unknown sequences in 15 species

TABLE III. ACCURACY OF THE CLASSIFIERS FOR SET OF 25 SPECIES IN THE KNOWN SET WITH VARIOUS % OF UNKNOWN SEQUENCES

% 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	54.20
70	52.50	54.35	53.80
80	50.99	53.04	52.42
90	49.38	51.39	50.73

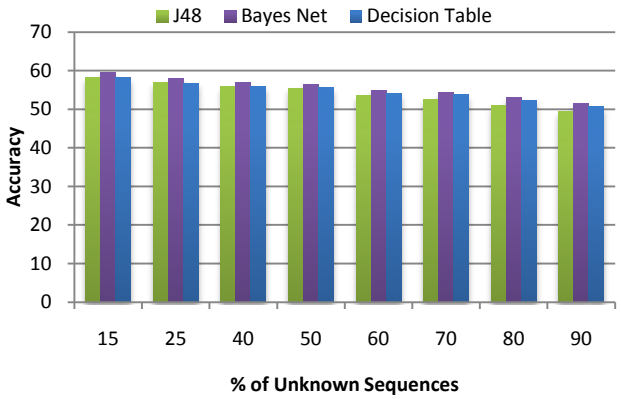

Figure 3. Performance of classifiers with different % of unknown sequences in 25 species

V. 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. We can also try to apply meta learners such as bagging, boosting and stacking to see how their performance scales.

VI. 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 article 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 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 much higher accuracy than the expected random guessing accuracy. Training and testing using approximately 50,000 instances just took 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.

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 Acids Research, 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," 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].