TAXONOMIC ASSIGNMENT OF GENE SEQUENCES USING HIDDEN MARKOV MODELS

By Huanhua Huang

A Thesis

Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Science

In Engineering

Northern Arizona University

August 2014

Approved:

J. Gregory Caporaso, Ph.D., Chair

Dieter Otte, Ph.D.

Talima Pearson, Ph.D.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

ABSTRACT

TAXONOMIC ASSIGNMENT OF GENE SEQUENCES

USING HIDDEN MARKOV MODELS

Huanhua Huang

Our ability to study communities of microorganisms has been vastly improved by the development of high-throughput DNA sequences. These technologies however can only sequence short fragments of organism's genomes at a time, which introduces many challenges in translating sequences results to biological insight. The field of bioinformatics has arisen in part to address these problems.

One bioinformatics problem is assigning a genetic sequence to a source organism. It is now common to use high-throughput, short-read sequencing technologies, such as the Illumina MiSeq, to sequence the 16S rRNA gene from a community of microorganisms. Researchers use this information to generate a profile of the different microbial organisms (i.e., the taxonomic composition) present in an environmental sample. There are a number of approaches for assigning taxonomy to genetic sequences, but all suffer from problems with accuracy. The methods that have been most widely used are pairwise alignment methods, like BLAST, UCLUST, and RTAX, and probability-based methods, such as RDP and MOTHUR. These methods can classify microbial sequences with high accuracy when sequences are long (e.g., thousand bases),

however accuracy decreases as sequences are shorter. Current high-throughout sequencing technologies generates sequences between about 150 and 500 bases in length.

In my thesis I have developed new software for assigning taxonomy to short DNA sequences using profile Hidden Markov Models (HMMs). HMMs have been applied in related areas, such as assigning biological functions to protein sequences, and I hypothesize that it might be useful for achieving high accuracy taxonomic assignments from 16S rRNA gene sequences. My method builds models of 16S rRNA sequences for different taxonomic groups (kingdom, phylum, class, order, family genus and species) using the Greengenes 16S rRNA database. Given a sequence with unknown taxonomic origin, my method searches each kingdom model to determine the most likely kingdom. It then searches all of the phyla within the highest scoring kingdom to determine the most likely phylum. This iterative process continues until the sequence cannot be assigned at a taxonomic level with a user-defined confidence level, or until a species-level assignment is made that meets the user-defined confidence level.

I next evaluated this method on both artificial and real microbial community data, with both qualitative and quantitative metrics of method performance. The evaluation results showed that in the qualitative analyses (specificity and sensitivity) my method is not as good as the previously existing methods. However, the accuracy in the quantitative analysis was better than some other pre-existing methods. This suggests that my current implementation is sensitive to false positives, but is better at classifying more sequences than the other methods.

I present my method, my evaluations, and suggestions for next steps that might improve the performance of my HMM-based taxonomic classifier.

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to all those who provided me the help to make my thesis possible.

I would like to thank to my research advisor, Professor Greg Caporaso. Without his guidance and inspiration, I couldn't have passed through this challenging time of learning bioinformatics. I thank him for his patience in answering the many questions I had through-out this project for his guidance on my project, and for his edits on my thesis. His support motivated me to continue this study.

I would also like to thank to Christopher Coffey for helping me learn to work with "monsoon", NAU's high-performance computing cluster. I thank Chris for his patience in helping me solve the problem of running thousand commands on "monsoon" to generate taxonomic Hidden Markov Models of 16S rRNA sequences.

Furthermore, I would like to thank to Laura Bohland for helping with revisions on my thesis, and for her patience while advising me on writing style in my thesis. I would also like to thank to my friends, Paige Williford and Tairee Fang, for helping me with grammar and writing style in my thesis.

In addition, I would like to thank my committee members, Professor Dieter Otte and Professor Talima Pearson for attending my project meetings and giving useful suggestions and different perspectives on my project. Also, thanks to John Chase, Giorgio Casaburi, Evan Bolyen, Katy Califf, and Crystal Hepp in Dr. Caporaso's laboratory who helped me answer my questions about computer programming, bioinformatics tools, and biology.

Finally, I would like to thank my parents, my younger sister, and my other close friends. They were always supportive and encouraging me with their best wishes.

TABLE OF CONTENTS

ABSTRACT ii
ACKNOWLEDGEMENTS iv
TABLE OF CONTENTS vi
LIST OF TABLES vii
LIST OF FIGURES viii
Introduction 1
Chapter 1 An Analysis of Microorganisms 3
1.1 The study communities of micro-organisms 3
1.2 Culture-dependent and culture-independent analysis of microbial communities 4
1.3 Using 16S rRNA to study communities of microorganisms 6
1.4 Limitations of using 16S rRNA for studying communities of micro-organisms 7
Chapter 2 Existing Methods for Assigning Taxonomy to DNA Sequences 9
2.1 An overview of pairwise alignment 9
2.2 Alignment-based approaches 11
2.3 The Naïve Bayes classifier 15
2.4 The RTAX classifier 17
Chapter 3 Taxonomic Classification of DNA Sequences using Profile Hidden Markov Models 19
3.1 An overview of Hidden Markov Model 19
3.2 Profile Hidden Markov Models 26
3.3 The method of assigning 16S rRNA sequences based on the profile Hidden Markov Models 29
Chapter 4 The Evaluation Framework 31
Chapter 5 Experiment Results 36
Chapter 6 Conclusions 43
References 45

LIST OF TABLES

Table 1 8 samples with precision, recall, F-measure, and correlation results when E-value is 100. 38
Table 2 The precision, recall, F-measure, and correlation results along with eight samples when E-value = 1. 39
Table 3 The precision, recall, F-measure, and correlation results along with 8 samples when the E-value = 1e-30 40
Table 4 The $M2$ with methods and parameters. 42

LIST OF FIGURES

Fig. 1 The gaps between two alignment sequences. 10
Fig. 2 An alternative way to align two sequences. 10
Fig. 3 The process of query words chosen from a list of words containing all possible words. 13
Fig. 4 Each query word corresponds to a group of sequences containing its query word. 13
Fig. 5 An example of a Hidden Markov Model. 21
Fig. 6 Multiple alignment DNA sequences. 26
Fig. 7 Profile Hidden Markov Model with match states. 27
Fig. 8 A Profile Hidden Markov Model with match and insert states. 27
Fig. 9 A Profile Hidden Markov Model with possible deletions. 28
Fig. 10 A Profile Hidden Markov Model with match and delete states. 28
Fig. 11 A complete structure of the Profile Hidden Markov Model. 28
Fig. 12 The database contains four different microorganisms. 30
Fig. 13 Seven clusters of the microorganisms based on their taxonomic level. 30
Fig. 14 A tree structure of the database. 30
Fig. 15 An example of a distance matrix and its PCoA plot. 33
Fig. 16 The precision results with different classifiers when the E-value of the hmmtax is 100. 38
Fig. 17 The recall results with different classifiers when the E-value of the hmmtax is 100. 38
Fig. 18 The F-measure results with different classifiers when the E-value of the hmmtax is 100. 38
Fig. 19 The correlation results with different classifiers when the E-value of the hmmtax is 100. 38
Fig. 20 The precision results with different classifiers when the E-value of the hmmtax is 1. 39
Fig. 21 The recall results with different classifiers when the E-value of the hmmtax is 1. 39
Fig. 22 The F-measure results with different classifiers when the E-value of the hmmtax is 1. 40
Fig. 23 The correlation results with different classifiers when the E-value of the hmmtax is 1. 40
Fig. 24 The precision results with different classifiers when the E-value of the hmmtax is 1e-30. 41
Fig. 25 The recall results with different classifiers when the E-value of the hmmtax is 1e-30. 41
Fig. 26 The F-measure results with different classifiers when the E-value of the hmmtax is 1e-30. 41
Fig. 27 The correlation results with different classifiers when the E-value of the hmmtax is 1e-30. 41

Introduction

"It is estimated that there are more than 10 million—perhaps 100 million—living species on Earth today." (Zvelebil & Baum, 2008). The vast majority of these species are microorganisms. They exist in the extremely cold polar ice cap to the high-temperature volcanoes, from the top of mountains to the Earth's crust, from the bodies of plants to those of animals. These microorganisms play an important role in the ecosystem. They connect nonliving components of an environment, like water or air with a great variety of living organisms.

When we observe these microorganisms, we see that even though each organism is different, they still have similar characteristics, such as appearances or functions. Based on Mendel's genetic theory, these characteristics are inherited from their parent organisms. That is, parent organisms pass down information containing characteristics to their offspring. This process is very different from other processes, such as the melting of ice and the burning of a candle, where the substances only are changed by their form and no new attributes are generated. However, living organisms constantly require energy to maintain their internal organizations to drive the complex system of chemical processes, which is specified by hereditary information.

This hereditary information exists in the form of DNA sequences. By using modern technologies, like Next-Generation Sequencing (NGS) (Liu, et al., 2012), DNA sequences can be extracted in high volume from microorganisms. Because of these advanced technologies, we can now collect DNA sequences from unknown microorganisms and further classify unknown microorganisms through the analysis of

DNA sequences. Then, we can know the relationships between these microorganisms by building a phylogenetic tree based on their taxonomic assignments. There are several current methods that can identify a classification for unknown microorganisms, such as the BLAST or RDP classifier. However, these often can only classify taxonomic levels down to the genus level. Also, these classifiers sometimes assign the wrong taxonomies at a certain taxonomic level, and sometimes they can only accurately achieve a higher-level classification of organism, for example the phylum level. To conquer these problems, a breakthrough method is proposed in this thesis, which assigns taxonomies to those unknown microorganisms based on the profile Hidden Markov Model. The data will show that this method can raise the accuracy of taxonomy assignment.

Chapter 1

Analysis of Microorganisms

1.1 Introducing microorganisms

Microorganisms, also known as microbes, are single cellular organisms that generally cannot be seen with the naked eye. Microbes span the three domains of life, the archaea, bacteria, and eukarya, are found all around our world including the air, soils, mountains, lakes, oceans, animals, and in and on human bodies. These microbes play a vital role in each of these ecosystems. For example, some microbes decompose dead organisms, animal waste, or plant litter to obtain their nutrients. During this process of decomposition, carbon, nitrogen, and phosphorous are recycled providing necessary nutrients for maintaining life in other species (Department of Health and Human Services, 2009).

Some microbes (Harley, 2009) exist at the bottom of the food chain to serve as food for other organisms. For example, in the Great Salt Lake several types of cyanobacteria (formerly known as blue-green algae) and *Dunaliella* (free-floating algae), are primary producers, generating their energy from sunlight. They become food for primary consumers, such as brine shrimps and brine flies. Waterbirds and shorebirds are secondary consumers and eat the primary consumers. Without the microbes, the waterbirds and shoebirds would not survive.

Some microbes (Biello, 2010) can have industrial roles, for example in pollution control, because they naturally break down metals, acids, salt, methane, and even radioactive waste into other, less problematic chemical structures. Other microbes

(Wijffels & Barbosa, 2010) are potential sources of renewable energy by converting animal fat into usable fuel. And microbes living in or on human body can have both positive and negative effects on our health. For instance, some bacteria that live in the human gut break down carbohydrates and provide vitamins, like K and B12. These nutrients that are important for our health are of bacterial origin (Department of Health and Human Services, 2009).

It is estimated that less than 1% of existing microorganisms have been cultured, or grown in the lab (Ward, Weller, & Bateson, 1990). For the remainder, we only known of their existence because we have observed their DNA sequences repeatedly using culture-independent approaches, including high-throughput DNA sequencing, which is the primary source of the gene sequences that I present on here.

1.2 Culture-dependent and culture-independent analysis of microbial communities

The traditional method of identifying microorganisms includes five steps, which are inoculation, incubation, isolation, inspection, and identification. Each of these procedures take place in the laboratory (Talaro, 1999).

First, microorganisms are collected from an environmental source, such as water, soil, or sewage, and injected into a container filled with nutrient media. The nutrient media is used to create a growth environment for the samples. This process is called inoculation. Then, containers are put into an incubator, which provides the proper growth temperature for hours, days, or weeks. During this time, ideally, microbes will grow and reproduce in the medium. This process is called culture. If the observation of an individual microbe is desired, the next step is to isolate cells from the microbial species

of interest from the others. A single cell will reproduce, creating a discrete mound of cells, which is called a colony. The colony then goes to the inspection stage in order to observe the growth characteristics, such as size, color, or texture which could be useful in identifying the organisms and biochemical tests. Finally, if sufficient information is obtained in the inspection stage, the taxonomy of microbes can be determined (Talaro, 1999).

Each of these can be very challenging. For example, isolation can be difficult or impossible because one species of microbes can have complex associations with other species of microbes in natural environment, including parasitism and mutualism. It may therefore be impossible to observe either or both species separately. Additionally, when attempting to grow a certain species of microbe in a laboratory setting, undesired microbes might be introduced by accident into the experiment, which can cause contamination and misidentification. This can occur because they are so small, so easily dispersed, and prevalent everywhere. Finally, it can be difficult to know what nutrients a microbe will require to grow. So designing the nutrient media can require large amounts of trial and error. These are only a few of the challenges that microbiologists face when attempting to grow microbes in isolated cultures. This culture-based methodology has given microbiologists a perspective with which to study species of organisms. However, the challenges cited above have led to another approach, culture-independent analysis, can be utilized to complement the culture-based methods (Talaro, 1999).

Culture-independent method, for studying or identifying microorganisms generally involve the analysis of DNA sequences from communities of microorganisms, so not requiring isolation. Currently, the most common procedure is to extract DNA from a

sample, then amplify a single "marker gene" by utilizing the polymerase chain reaction (PCR). PCR is a method that targets a specific DNA sequence, and then can create billions of copies of it in several hours. This technology has been a major breakthrough in identifying unknown microbes. Scientists can now identify the microbes present in a sample at the same time, without the complexities of culture (Pogacic, Kelava, Zamberlin, Dolencic-Spehar, & Samarzija, 2010).

1.3 Using 16S rRNA to study communities of microorganisms

In culture-based identification of microorganisms, scientists identified microorganisms using their phenotypic characteristics, such as shape, size, color, and behavior. However, with the invention of PCR and DNA sequencing technologies, scientists can now isolate and sequence universally conserved genes, such as the 16S rRNA gene, 18S rRNA gene, and 23S rRNA gene, from different microorganisms and use the sequence of these genes, which differ between microbial species, to identify which organisms a specific gene is derived from. Among these genes, the 16S rRNA gene has most widely used because it contains highly conserved regions, meaning that regions of the gene sequence are identical or nearly identical across microbial species, which is necessary for PCR. In addition, this gene is universally found in the bacteria and archaea, which make up the majority of microbial taxa. Finally, in addition to containing highly conserved regions, it includes highly variable regions, which allow for the identification of different organisms. Therefore, researchers started using the 16S rRNA sequence to classify the taxonomy of microorganisms found in a sample, but the best approach or approaches for performing this classification is still an area of active research.

This approach is very powerful, as it can classify organisms that cannot be cultured and therefore makes the remaining 99% of microbial taxa accessible to researchers.

1.4 Limitations of using 16S rRNA for studying communities of micro-organisms

Although there are many advantages to using the 16S rRNA sequences mentioned above, there are no widely accepted approaches for performing taxonomic assignment of 16S rRNA sequences. Additionally, making a taxonomic assignment requires the existence of a 16S rRNA database, containing 16S sequences and their taxonomic assignments. The sequence database is used to search for sequences that are similar to a microbial "query" sequences, for example one identified from the environment. But the assignments can only be as good as the database, and there is uncertainty in how accurate each sequence in the database is, how accurate each taxonomic assignment in the database is, and how comprehensive in terms of the taxonomic coverage the database is. Some nucleotides in a database or query sequence can be incorrect due to technical errors in obtaining them, and some microbial sequences will not be included in the database because we still have not surveyed the full extent of microbial life, and it's unlikely that we will anytime soon. These situations will limit the accuracy of identification of microorganism from their DNA sequences, regardless of the method used to perform the identification (Wu, Lau, Teng, Tse, & Yuen, 2008).

When it comes to the interpretation of the sequence data, several parameters need to be considered before searching a database. These parameters are the length and the quality of the sequence data, the choice of an appropriate software package for performing the search, and settings for any free-parameters for that software package,

including software-specific similarity thresholds for considering matches to the database for final sequence assignment.

Researchers found that the accuracy of taxonomic classification obtained from a classification method can vary with features including length of a query sequence, and regions of the gene sequence used for classification (since most high-throughout sequencing methods cannot sequence full-length genes) (Liu, DeSantis, Andersen, & Knight, 2008). In addition, different classified methods differ in their ability to infer the taxonomic origin from the same length of a query sequence, so some taxonomic assigners may work better on some gene fragments, while others work better on other gene fragments. For example, BLAST, RDP, and the Greengens Online classifier are similarity–search based methods, and Fitch, FitchAndBack and LCA are phylogenetic tree-based methods. The tree-based methods were found to have better overall accuracy than similarity-search based methods when searching full-length gene sequences. When the performance of classifiers was compared based on gene fragments of 100bases, 250 bases, 400bases, the tree-based methods performed less well than the similarity search methods for some (but not all) of the sequenced gene regions. In the next chapter I will introduce some of these methods for taxonomic assignment in more detail.

Chapter 2

Existing Methods for Assigning Taxonomy to DNA Sequences

As mentioned previously, most of the microorganisms on earth have yet to be discovered, so we do not have a comprehensive database that maps DNA sequences to taxonomic origin. Therefore to give a taxonomic classification to an unknown DNA sequence, bioinformaticians have developed methods to classify a newly discovered genomic sequence by comparing it to sequences from known microorganisms (Koonin & Galperin, 2003). When working with a single genetic locus (i.e., region of the genome) that is shared across organisms, the similarity and differences between an unknown sequence and known sequences can help identify what type of organism a sequence may have come from. Therefore, when comparing unknown sequences against a database, the greater the similarities between two sequences, the greater the chance that they are derived from similar microorganisms. Currently, there are a variety of software tools that allow for searching unknown sequences against a database of taxonomically annotated reference sequences to determine their most likely taxonomic origin. In this chapter I will discuss the methods that have been most widely used for taxonomic assignment of microbial sequences.

2.1 An overview of pairwise alignment

Pairwise alignment is a way to identify similarities between two DNA sequences by lining up bases that are hypothesized to be homologous. If the two DNA sequences have

the same lengths derived from a common ancestral sequence, identical base pairs should exist between the two DNA sequences. However, the process of evolution typically introduces changes over many eons in the form of mutations, which cause differences in present-day sequences that are derived from a common ancestor. That is, over time, some nucleotides in are substituted with other nucleotides, some nucleotides are inserted into the original sequence, and some nucleotides are deleted from the original sequence. These mutations result in different lengths and compositions of sequences that derive from a common ancestor. When we align these sequences containing insertions and deletions relative to one another, alignment algorithms insert gaps into the shorter sequences to make to model these changes, as in Fig. 1 (Zvelebil & Baum, 2008).

A T C G G G T
A – C – – G T

Fig. 1 An alignment of two sequences illustrates insertion/deletion events that result in gaps in the second of the aligned sequences.

Since there can be many different ways to align any two sequences, the next issue is how to objectively estimate the quality of an alignment to determine if one alignment is better than another. For example, the above alignment sequences can also be aligned as shown in Fig. 2.

A T C G G G T
A – C – G – T

Fig. 2 An alternative way to align the two sequences in Fig 1.

To address this problem, alignment algorithms score each match, mismatch, insertion, and deletion in the alignment, and use that information to find the optimally scoring alignment. For instance, if a base pair are identical, a score of +5 might be assigned to the

base pair. If the base pair are mismatched, a score of -4 might be assigned to this base pair. The insertion and extension of gaps are also scored. Typically, insertion of a new gap might be assigned to a score of -10 and any adjacent gaps might be assigned to a score of -2. In other words, open a new gap incurs a large penalty, and extending an existing gap incurs a smaller penalty. The scores assigned to an alignment will depend on the scoring scheme that is implemented.

There are two widely used algorithms for computing and scoring alignments. One is called the Needleman-Wunsch algorithm (Needleman & Wunsch, 1970) and the other is called the Smith-Waterman algorithm (Smith & Waterman, 1981). The Needleman-Wunsch algorithm considers the whole length of each sequence and optimizes the alignment between the two sequences. This is called global alignment. On the contrary, the Smith-Waterman algorithm is a variation of the Needleman-Wunsch algorithm which locates only a segment of the sequences that have high similarity and aligns them. This is called local alignment. Since it is very difficult to obtain the correct alignment of low similarity regions when attempting to align two distantly related sequences using the Needleman-Wunsch algorithm, and because current sequence technologies typically only give us short sequence fragments, the Smith-Waterman algorithm has become more widely in database searching (as is done for taxonomic assignment).

2.2 Alignment-based approaches

"Basic Local Alignment Search Tool", or BLAST, is a tool that uses local alignment to identify pairwise sequence similarity when searching for homologous nucleotide or amino acid sequences in a database (Altschul, Gish, Miller, Myers, & Lipman, 1990). To

summarize, the BLAST algorithm uses local alignment to identify known database sequences that are similar to a query sequence by locating small regions (usually less than 10 bases) in the database that match corresponding subsequences within the query sequence, and align these high scoring segment pairs. In order for one of these segment pairs to be included in the full alignment, their similarity score must be above a certain threshold score, defined as *T*. *T* is an arbitrary number that is chosen by the user. From simulations of different chosen scores of *T*, when the length of the sub-sequence extracted from a query sequence was three and the score of *T* was eleven, BLAST could find 99% segment pairs that had maximum similarity scores determined by local alignment similarity calculation methods. All segments that satisfy the *T* threshold are extended until their score does not continue to increase (the query sequence no longer matches the database sequence). The segment pair with the highest similarity score is termed the maximum segment pair (MSP) and its similarity score is defined as S. S is the cutoff similarity score for an MSP.

The details of the three steps that make up the BLAST algorithm are as follows. First, a word list containing all possible subsequences with the length w is created from the query sequence. W is an arbitrarily chosen number, although three is often used. The similarity between these words and the original full-length query sequence are then calculated. If the score of the alignment is greater than T, the word will be collected into a new query word list. Otherwise, the subsequence will be discarded (Fig. 3). The words in the new query word list are called *w*-mers. Second, the *w*-mers in the list are scanned through the database to identify which sequences in the database have the potential to become MSPs, as described above. The sequences from the word list containing the most

matching *w*-mers are selected as in Fig. 4. These represent the sequences most likely to align well to the query sequence. Third, the *w*-mers from the selected sequences are extended with pairwise alignment against the query sequence until they are MSPs. This step is repeated until segment pairs with a score greater than or equal to *S* are found. Finally, the output lists all MSPs that have scores of at least *S*. These are the segment pairs with the largest values of S, and are expected to be the database sequences that are most similar to the query sequence. When BLAST is used to assign taxonomy to a sequence, the taxonomic of the best database match is assigned to the query sequence.

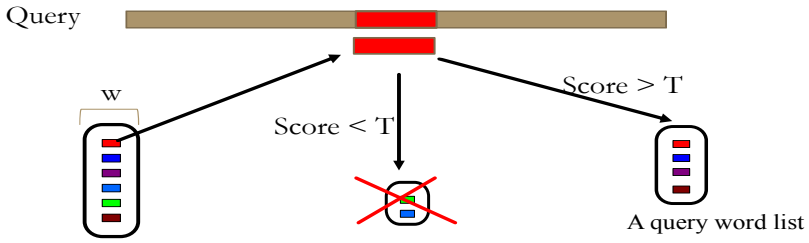

Fig. 3 The process of query words chosen from a list of words containing all possible words.

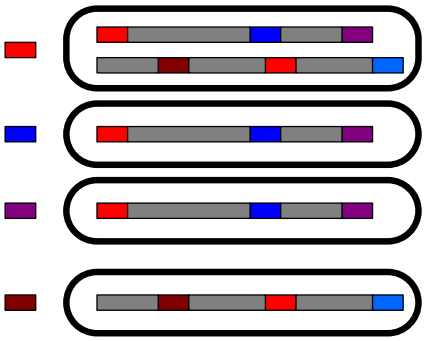

Fig. 4 Each query word corresponds to a group of sequences containing its query word.

The BLAST algorithm can be very time consuming, since each sequence in the database is scanned several times depending upon the number of query words. The uclust algorithm (Edgar, 2010) was developed to expedite this process.

The process of matching sequences with uclust is similar to using BLAST. Each sequence in a database is scanned to see if it shares short words with a query sequence. However, the difference between BLAST and uclust lies in the method of the alignment after sequences with similar word composition are identified. BLAST uses local alignment to match subsequences from query words to known sequences in a searched database, and then calculate their similarity. The uclust algorithm uses global alignment to search for the target matching sequences in a database with whole query sequences. When calculating the similarity between the query and target sequences, the similarity defined by uclust is defined as "identity", which is the number of matches in a global alignment divided by the length of the shorter sequence. If the identity is over a certain arbitrary threshold, then the target sequence will be "*accepted*". Otherwise, it will be "*rejected*".

When using uclust to search a query sequence against a database, uclust does not compare the query sequence to the entire database, but instead to representative sequences collected from different clusters at a certain chosen similarity score. These database sequences are then sorted based on the number of unique words of any length (typically a minimum length of three) in common between the query sequence and the ith database sequence. When starting to search for target sequences in the database, the database sequences are compared to the query sequence in order of this list. If the query sequence is matched to the first database sequence in the list, this query sequence will be assigned to the corresponding cluster, and uclust will stop searching for the next possible match sequence. However, if the query sequence is not matched to the first database sequence in the list, that database sequence will be rejected. The uclust algorithm will

continue to search against the next database sequences, in order of the list, until a target sequence is found. Once the number of "*rejected*" is over a certain number, for example, 8, the remaining sequences in the database will no longer be searched by uclust, as the words further down the list have fewer unique words in common between the query and the database sequence. This algorithm can dramatically decrease the run time for searching a database for sequences similar to a query sequence, as far fewer sequences in the database are searched when compared with the BLAST algorithm.

2.3 The Naïve Bayes classifier

The Naïve Bayes classifier is a probabilistic classifier, which uses Bayes' theorem to classify unknown microbial marker gene sequences. The classifier identifies microbial genomic sequences from a database belonging to a certain taxonomic level, based on probability scores using Bayes Theorem. The widely used Naïve Bayes classifiers are RDP (Ribosomal Database Project) classifier (Wang, Garrity, Tiedje, & Cole, 2007) and MOTHUR (Schloss, et al., 2009). The Ribosomal Database Project is a software package, which provides the data, tools, and services related to the 16S rRNA sequences. The RDP classifier used in the RDP package assigns taxonomies to microbial sequences. MOTHUR, a software package for microbial community analysis written in C++, incorporates pre-existing algorithms with some additional features, such as visualization tools and an increase in efficiency.

For the Naïve Bayes method, the sequences in the database are first clustered based on sequence similarity at different taxonomic levels, for instance, kingdom or phylum. A word list (W) is then created from a set of words ($w_1, w_2, \ldots, w_n$) that are extracted from

query sequences, as in the BLAST and UCLUST algorithms. The default length of each word is 8 bases. $n(w_i)$ is defined as the number of sequences containing a word i. If the total number of sequences is N, the probability that a sequence that contains a word i is

$$P_i = \frac{n(w_i) + 0.5}{N + 1},$$

where the values of 1 in the denominator and 0.5 in the numerator are used to make the value of P_i between 0 and 1. Given a query sequence S, the probability that this sequence is a member of one of the clusters, G, at a given taxonomic level, is defined as

$$P(G|S) = \frac{P(S \cap G)}{P(S)}$$

Based on Bayes' theorem, the probability of the intersection of the query sequence S and the cluster G, which is $P(S \cap G)$ in the above equation, can be replaced with $P(S|G) \times P(G)$. The above equation then becomes

$$P(G|S) = \frac{P(S|G) \times P(G)}{P(S)},$$

where $P(G)$ represents a prior probability of a sequence being a member of a cluster and $P(S)$ is an overall probability of the query sequence found in the database. $P(S|G)$ cannot be calculated directly, however it can be estimated. The query sequence S contains a set of words $V = \{v_1, v_2, \dots, v_f\} \subseteq W$, If the dependence between each word is ignored here, the probability of each word in a word set V that is a member of G at the same time can be calculated with $\prod_{i=1,\dots,f} P(v_i|G)$. For the cluster G with M sequences, let $m(w_i)$ represent the number of sequences in the cluster G containing a word w_i. The probability of a word w_i is a member of the genus cluster G, and formula is

$$P(w_i|G) = \frac{m(w_i) + P_i}{M + 1}$$

Assuming that $P(S)$ and $P(G)$ have equal probabilities over all taxa, they can be ignored in the equation $P(G|S)$. Since the values of $P(S)$ and $P(G)$ do not change, the Naïve Bayes formula is reduced to $P(G|S) = P(S|G)$, which determines the score of each query sequence in order to identify the cluster at a taxonomic level. Therefore, to identify the taxonomy of a given query sequence, each taxon in the taxonomy is determined by a cluster with the highest probability score at each taxonomic level.

2.4 The RTAX classifier

The RTAX classifier (Soergel, Dey, Knight, & Brenner, 2012) has slight differences from the uclust method. Uclust uses only one short read extracted from a 16S rRNA gene sequence. However, RTAX uses two short reads at either end of a 16S rRNA gene sequence to perform taxonomic classification, as are generated in paired end sequence (a common output of current DNA sequences). The advantage is that more sequences can be used in taxonomic assignment, so that when ambiguous information (such as an equal scoring match to database sequences with different taxonomies) occurred there is more information available to make an assignment. When RTAX searches for a database with the two short reads, each short read is queried against the database using uclust algorithm, as described above. If a query sequence matches to one of sequences in a reference database, it is called a hit. Each hit is based on the minimum percent identity (%id) threshold to see if a hit is accepted or rejected and to be outputted as a result.

RTAX classifier uses different %id values as thresholds to iteratively filter database sequences with different %id values. First, the RTAX uses a stringent %id threshold to

find the good matching sequences. The sequences, where %id values do not reach the stringent %id threshold, are collected into another list. Then, the uclust runs this list with a lesser %id threshold to generate a list of hits. Those sequences that do not have good matches form a list and are ran through uclust again until either a good match is found or the lowest %id threshold is exceeded. Finally, when hits are found, database matches are outputted from each job. Then, RTAX interprets the results by intersecting the results from the uclust jobs. The matching sequences found in the match lists generated from each job are compared between each other. If the identical matching sequences are found in both of match lists, the average %id is computed from both identical matching sequences. The target sequences are selected when the average %id of matching sequences is less than 0.5% compared to the highest %id of the matching sequence.

RTAX using two %id threshold, which one is the highest threshold and one is the lowest threshold, is to find some database sequences that are not easily to be matched with the query sequences by only using a high similarity threshold. If the stringent value is too high, it could make some distant match sequences that can cause information to be classified at a higher taxonomic level and could be filtered out from the hit list. If the %id value is set too low, some imperfect match sequences could pass to the hit list. When taxonomy coverage in a database is biased towards some taxa, these imperfect match sequences will be regarded as not correctly identified sequences and then cause a decrease in the accuracy of taxonomic classification. Therefore, two identity thresholds ensure no distant related match sequences can be excluded, also ensure imperfect sequences can be filtered out from match lists.

Chapter 3

Taxonomic Classification of DNA Sequences using Profile Hidden Markov Models

A profile Hidden Markov Model is a Hidden Markov Model that is built from a family of biological sequences. Here, I apply it in microbial community analyses to assign taxonomy to microbial sequences. This approach does not use the traditional method of assigning taxonomies by using pairwise alignment. Instead, it adopts a probabilistic method, profile Hidden Markov Models, to build and search a database for homologs of DNA sequences. Profile Hidden Markov Models are usually used to identify the relationships between each individual biological sequence in a family, since they have the same or related functions. The main reason for using the profile Hidden Markov Models is that they can capture the position-specific information for nucleotide sequences. Currently, there is a software tool called HMMER (Eddy & Wheeler, 2013) that can perform the profile analysis for multiple alignment sequences using Hidden Markov Models, which makes it possible to search query sequences against the database. To further understand what the Hidden Markov Models are, the next several sections will introduce an overview and discuss how the profile Hidden Markov models are applied to the database.

3.1 An overview of Hidden Markov Model

The Hidden Markov Models (HMMs) (Rabiner & Juang, 1986) are a statistical method capable of generating hidden state transition paths based on transitional

probabilities as they relate to observable output. For the purposes of this thesis, the HMM parameters are defined as:

- $T =$ the length of an observed sequence.
- $N =$ the number of states in a model.
- $M =$ the number of observed symbols in a model.
- $\bar{S} = \{s_1, s_2, \dots, s_N\}$, which is a set of all possible states.
- $\bar{Q} = \{q_1, q_2, \dots, q_T\}$ is a sequence of state transitions and $q_T \in \bar{S}$.
- $\bar{X} = \{x_1, x_2, \dots, x_T\}$ is a collection of observed outputs.
- $\bar{V} = \{v_1, v_2, \dots, v_M\}$ is a collection of possible symbol observations.
- $\Theta = \{\pi, A, B\}$ is a set of parameters.
  - π: a vector of the *prior probabilities*. This is the probability that each state s_i is the first state of a state of a sequence.
  - A: a matrix of *transition probabilities* between state s_i and state s_j is $\{a_{i,j}\}$, where $\{a_{i,j}\} = P(q_{t+1} = s_j | q_t = s_i)$.
  - B: a matrix of *emission probabilities* that characterize the likelihood of an observation x is $\{b_{i,j}\}$, if the model is in state s_j, where $\{b_{i,j}\} = P(x_t = v_k | q_t = s_j)$.

For example, if a DNA sequence is sequenced from an organism, each time only one nucleotide is read by the DNA sequencer, and finally the observer will see the whole DNA sequence. In fact, the process of determining each nucleotide in a DNA sequence can be described as in Fig. 5. The states of the HMMs are s_1, s_2, s_3, and s_4. The observed output of each state is {A, T, C, G}. The transition probabilities and emission probabilities are as labeled:

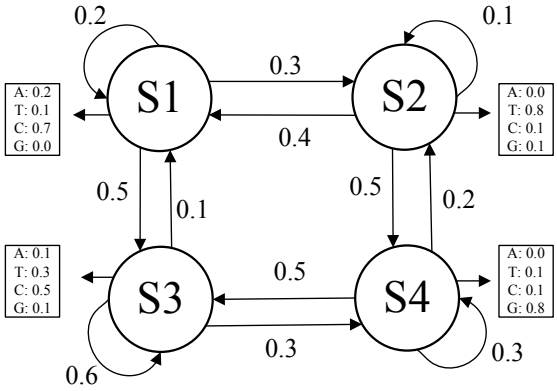

Fig. 5 An example of a Hidden Markov Model.

There are three key problems that will be considered in real world applications when using the Hidden Markov Model. These three problems are:

1. What is the probability of the observed outputs $x_1, x_2, \dots, x_T$ given the model Θ? That is, what is $P(x_1, x_2, \dots, x_T|\Theta)$?
2. Given the observed outputs $x_1, x_2, \dots, x_n$, which sequence of states has the largest probability? That is, what is $P(q_1, q_2, \dots, q_T|x_1, x_2, \dots, x_T, \Theta)$?
3. How do we adjust the parameters of a HMM model, such as $\pi, A, or\ B$ to maximize $P(x_1, x_2, \dots, x_T|\Theta)$?

For the first problem, the most straightforward way to calculate the $P(x_1, x_2, \dots, x_T|\Theta)$ is to find the probability of every possible state that would generate the observed outputs $x_1, x_2, \dots, x_n$, and determine the sum of these probabilities. Therefore, the likelihood of an observed sequence for a given model can be described by the summation of joint probability of an observed output X and a path Q over every possible Q as follows:

$$P(X|\Theta) = \sum_{every\ possible\ Q} p(X, Q|\Theta) \quad (1)$$

The joint probability of X and Q can be separated into a product of two quantities defined as follows:

$$p(X, Q|\Theta) = p(X|Q, \Theta) \times p(Q|\Theta) \quad (2)$$

For a given model and every fixed state sequence Q = $q_1, q_2, \dots, q_T$, the probability of the observed sequence is $p(X|Q, \Theta)$, where

$$p(X|Q, \Theta) = \prod_{t=1}^{T} p(x_t = v_k|q_t = s_i, \Theta) = b_{q_1,x_1} \cdot b_{q_2,x_2} \cdot \cdots \cdot b_{q_T,x_T} \quad (3)$$

The probability of each fixed state sequence for a given model is $p(Q|\Theta)$, where

$$p(Q|\Theta) = \prod_{t=1}^{T} P(q_{t+1} = s_j|q_t = s_i) = \pi_{q_1} \cdot a_{q_1,q_2} \cdot a_{q_2,q_3} \cdot \cdots \cdot a_{q_{T-1},q_T} \quad (4)$$

Equations (2), (3), and (4) can be substituted into the first equation (1), where

$$P(X|\Theta) = \sum_{q_1 q_2, \cdots, q_T} \pi_{q_1} \cdot b_{q_1,x_1} \cdot a_{q_1,q_2} \cdot b_{q_2,x_2} \cdot a_{q_2,q_3} \cdot \cdots \cdot a_{q_{T-1},q_T} \cdot b_{q_T,x_T} \quad (5)$$

The above equation (5) can be interpreted at the initial time (t=1) where we are in the state q_1 with the probability π_{q_1}, and generate the symbol x_1 with the probability b_{q_1,x_1}. Then, we make transitions from the state q_1 to the state q_2 with the probability a_{q_1,q_2} and generate the symbol x_2 with the probability b_{q_2,x_2}. After that, the next several similar transitions will be continued until the last state q_T is reached, which will generate the symbol x_T with the probability b_{q_T,x_T}.

When considering the computation time of the equation (5), if we calculate it by using the direct way of finding the probability of the observed sequence for each given path of state sequences, it will have the order of $2TN^T$ time complexity (($2T - 1)N^T$ multiplications and $(N^T - 1)$ additions), which will result in longer running time. To overcome this time issue, a forward algorithm will be applied instead. This dynamic programming algorithm breaks the problem down into several sub-problems, and then solves each sub-problem in order. The advantage of this is to store the solutions of each

sub problem into a memory. If the repeated sub-problems need to be solved the next time, the solutions already stored in the memory can be accessed. Therefore, forward procedure calculates the probability of the partial observed sequence (until time t) and the state q_i at time t for a given model, which is defined as:

$$\alpha_t(i) = p(x_{1,}x_{2,} \dots, x_{t,}q_t = s_i|\Theta) \tag{6}$$

$\alpha_t(i)$ can be solved by several steps, which are as follows:

1. Initially,

$$\alpha_1(i) = \pi_{q_i} \cdot b_{q_i,x_1}, \quad 1 \leq i \leq N \tag{7}$$

2. Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) \cdot a_{i,j}\right] \cdot b_{q_j,x_{t+1}}, \quad 1 \leq j \leq N \; and \; t = 1, \dots, T-1 \tag{8}$$

3. Finally,

$$p(X|\,\Theta) = \sum_{i=1}^{N} \alpha_T(i) \tag{9}$$

The first step initializes the probability of the observed sequence and state q_i at time t=1 ($1 \leq i \leq N$). After the initialization, the probability $\alpha_{t+1}(j)$ that the next state at time t+1 generates the observed symbol x will be obtained by summing every possible transition from state i to state j with the previous accompanying partial observed sequence and then multiplying them with probability $b_{q_j,x_{t+1}}$. This process will be repeated until time T. Finally, the sum of every variable $\alpha_T(i)$ will get $p(X|\,\Theta)$. If we examine the computation time, the order of time complexity of the algorithm is N^2T. This is much lower than the order of $2TN^T$ time complexity.

To answer problem 2, which attempts to find the optimal state sequences with respect to an observed sequence and a model, the highest probability of a single best path along

the observed sequence ending in state s_i at time t for a given model will be calculated. This can be expressed as

$$\delta_t(i) = max_{q_1,\dots,q_{t-1}} p(q_1 q_2 \dots q_{t-1}, q_t = s_i, x_1 x_2 \dots x_t | \Theta)$$

To solve this equation, the *Viterbi Algorithm,* which is similar to the forward method, is a good choice to recursively find the state sequences at time t that have the highest probability. Besides, another variable is needed to keep track of the best path ending in state s_i is $\varphi_t(i) = argmax_{q_1,\dots,q_{t-1}} p(q_1 q_2 \dots q_{t-1}, q_t = s_i, x_1 x_2 \dots x_t | \Theta)$. That is, $\varphi_t(i)$ can tell us which state at time $(t-1)$ will result in the highest probability of $\delta_t(i)$ at time t. Therefore, by applying the *Viterbi Algorithm,* $\delta_t(i)$ will be broke down into four parts:

1. Initially:

$$\delta_1(i) = \pi_i \cdot b_{i,x_1}, \quad i = 1, \dots, N$$

2. Induction:

$$\delta_t(j) = max_{1 \le i \le N} \left[\delta_{t-1}(i) \cdot a_{ij}\right] \cdot b_{j,x_t}, \quad 2 \le t \le T \text{ and } 1 \le j \le N$$

$$\varphi_t(j) = argmax_{1 \le i \le N} \left[\delta_{t-1}(i) \cdot a_{ij}\right], \quad 2 \le t \le T \text{ and } 1 \le j \le N$$

3. Termination:

$$p^*(X|\Theta) = max_{1 \le i \le N} \ \delta_T(i)$$

$$q_T{}^* = max_{1 \le i \le N} \ \delta_T(i)$$

4. Backtracking:

$$Q^* = \{q_1{}^*, \dots, q_T{}^*\} \ so \ that \ q_t{}^* = \varphi_{t+1}(q_{t+1}{}^*), \quad t = T-1, T-2, \dots, 1$$

The third problem, mentioned above, is obtaining the best probability of observed sequences given an HMM model and the associated parameters. The best parameters can be determined through a procedure that makes iterative adjustments. Among the current

iterative procedures, the Baum-Welch method will be discussed here. When re-estimating HMM parameters, several variables must be defined. The first variable is $\gamma_t(i)$, which is the probability of being in state s_i given the observed sequence and the model. It can be defined as

$$\gamma_t(i) = p(q_t = s_i|X, \Theta)$$

The second variable, $\delta_t(i,j)$, is the probability of being in state s_i at time t and making a transition to state s_j at time $t+1$ for a given observed sequence. It can be defined as

$$\delta_t(i,j) = p(q_t = s_i, q_{t+1} = s_j|X, \Theta)$$

To create a relationship between $\gamma_t(i)$ and $\delta_t(i,j)$, we sum $\delta_t(i,j)$ over j, which is defined as

$$\gamma_t(i) = \sum_{j=1}^{N} \delta_t(i,j)$$

Since the Baum-Welch method is an iterative procedure, the initial rough approximation parameters in the model are π, a_{ij}, b_{ix_j}, respectively, and we define the initial model as Θ. For the next several iterations, each parameter will be re-estimated by using the formulas, which are:

1. $\bar{\pi}_i = \gamma_1(i), \qquad 1 \le i \le N$
2. $\bar{a}_{ij} = \sum_{t=1}^{T} \delta_t(i,j) \,/\, \sum_{t=1}^{T} \gamma_t(i)$
3. $\bar{b}_{j,x_j=k} = \sum_{t=1,x_j=k}^{T} \gamma_t(j) \,/\, \sum_{t=1}^{T} \gamma_t(j)$

In the above formulas, $\bar{\pi}_i$ is the probability of being in state s_i at time $t = 1$. $\bar{a}_{ij}$ is the ratio of the expected number of transitions from state s_i to state s_j divided by the expected number of transitions from s_i. $\bar{b}_{j,x_j=k}$ is the ratio of the expected number of

times that s_j is visited and the observed symbol is k, divided by the expected number of times that s_j is visited. Given these re-estimated parameters, the new estimated model is defined as $\bar{\Theta}$. To improve the probability of the observed sequence from a given model, $\bar{\Theta}$ will be replaced with Θ to see whether $p(X|\bar{\Theta})$ will be greater than $p(X|\Theta)$. If $p(X|\bar{\Theta}) > p(X|\Theta)$, the observed sequence is most likely to be produced by the model $\bar{\Theta}$ instead of the model Θ. This process will be repeated until the local maximum point value of $p(X|\bar{\Theta})$ is reached.

3.2 Profile Hidden Markov Models

Hidden Markov Models are built for one sequence. However, profile Hidden Markov Models are built from a set of family sequences and are used to model what a family of sequences looks like. When building a HMM, these sequences must be in alignment with each other first in order to identify relationships among different sequences in a family. Fig. 6 is an example of multiple alignment of sequences.

ACAA
TCAA
ACAC
AGAA

Fig. 6 Multiple alignment of DNA sequences.

The profile Hidden Markov Models for this alignment will be built with one "match" state for each column, separated by the transitions of probability 1. Each match state M_i will emit one nucleotide with the emission probability, which comes from the number of occurrences of the nucleotide in the corresponding column. The "dummy" states of

"begin" and "end" will be added into the model with no output symbols. This profile HMM is shown in Fig. 7.

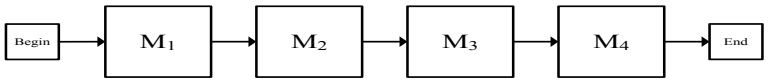

Fig. 7 Profile Hidden Markov Model with match states.

However, during evolution, sequences in a family diverge from each other, resulting in gaps. Insertions and deletions are types of gaps that occur in the sequences when doing the alignments. To deal with these gaps, the cases of insertions and deletions must be discussed separately. For the insertions case, since some portions of sequences do not match with the above model, additional states called "insert" states will be introduced in the above model. The model with insertion states I_i is shown in Fig. 8. The output probabilities from the insert state is set to the background probabilities, which means that the number of symbol K generated by the insert state i is divided by the number of possible symbols generated by the insert state I_i. Similarly, there will be a transition from the match state M_i to the insert state I_i, a loop transition to itself, and a transition from the insert state I_i to the match state M_j.

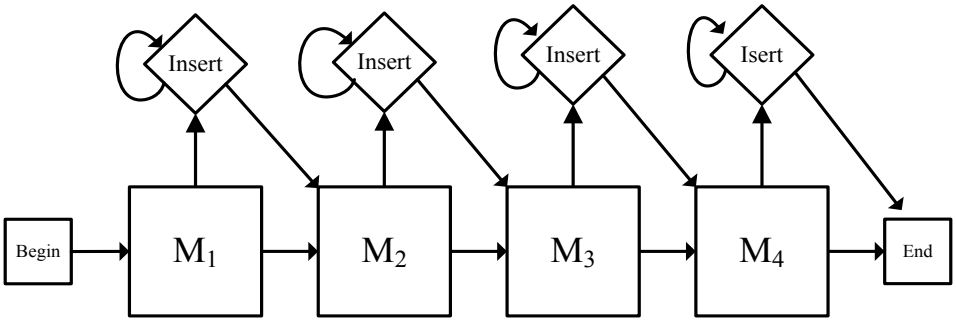

Fig. 8 A Profile Hidden Markov Model with match and insert states.

For the deletion case, since some DNA sequences are missing, many transitions will be introduced in the model, like in the Fig. 9. However, a symbol is not emitted when there is a transition from one state to another, and the transition will skip the middle states because some nucleotides are deleted during the evolution. In order to solve this problem, "delete" states with no emission probabilities will be introduced in the model, like in the Fig. 10. A complete structure of HMM with insertion and deletion states is shown in Fig. 11.

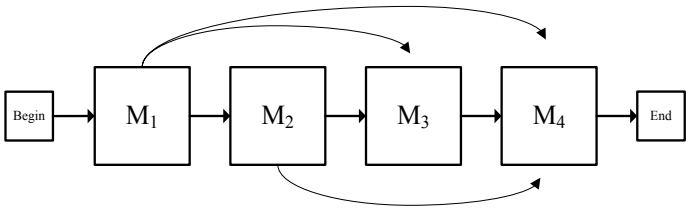

Fig. 9 A Profile Hidden Markov Model with possible deletions.

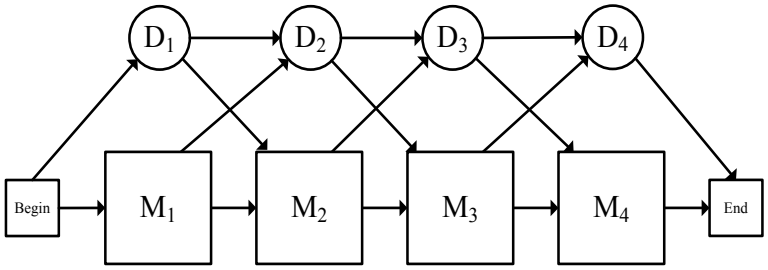

Fig. 10 A Profile Hidden Markov Model with match and delete states.

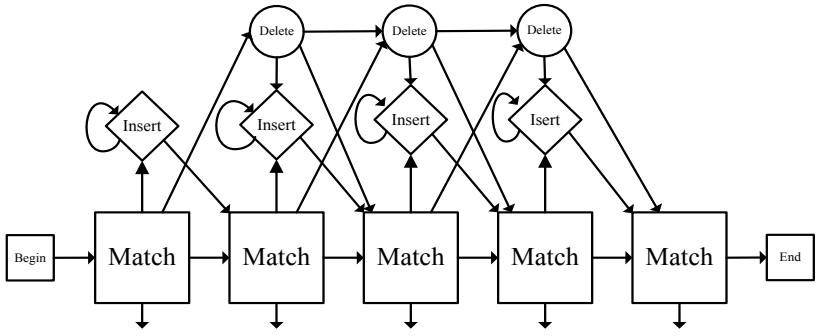

Fig. 11 A complete structure of the Profile Hidden Markov Model.

3.3 The method of assigning 16S rRNA sequences based on the profile Hidden Markov Models

The data obtained for the study is from the Greengenes database, where chimeric sequences found in public databases, such as NCBI, are filtered out. This insures that each DNA sequence will have accurate taxonomy assignments. (DeSantis, et al., 2006). The most recent Greengenes database was released in May, 2013 and contains 1,262,986 16S rRNA sequences belonging to Archaea or Bacteria with 203,452 99% operational taxonomy units(OTUs) and 99,322 97% OTUs (Caporaso, et al., n.d.). This experiment utilized a cluster of sequences at 97% identity from the Greengenes database.

To obtain the best taxonomy assignment at each taxonomic level, when a query sequence is compared against the Greengene database, database sequences are separated into different clusters that are each analyzed by the Hidden Markov Model. For example, if only four different microorganisms contained in the black block compose the database (Fig. 12), they first will be separated and grouped by taxonomic level, such as kingdom and phylum. (Fig. 13). Therefore, some groups of microorganisms belong to Archaea or Bacteria on the kingdom level, some groups belong to DHVE, pMC2A15, AC1 or BH1 on the phylum level, and some groups belong to LC-1 on the class level. Next, these groups will be connected based on their taxonomic relationships to build a tree structure of the database (Fig. 14). Finally, the HMMs will be applied to each group. Now, if an unknown query sequence is searched against the tree structure of the database, the query sequence will go to the first level of the two HMMs. Each HMM can find the most likely path and return an E-value to the query sequence. The E-value represents the expected number of sequences that will produce the same or a better score by chance, given some database size. If one of the E-values is smaller than the other, the query sequence will

descend from the HMM with the smaller E-value to the next phylum level of the HMMs and repeat the search for the best path process to the last level. The query sequence will keep repeating the same process until it arrives at the species level, then the database will return the taxon at each level, which goes from the species level to the kingdom level.

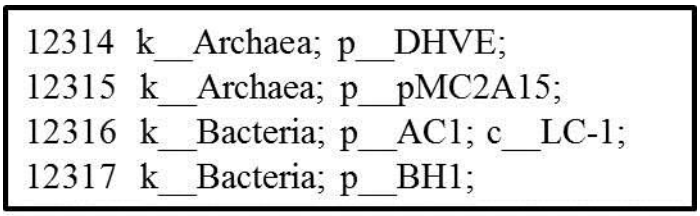

Fig. 12 The database contains four different microorganisms.

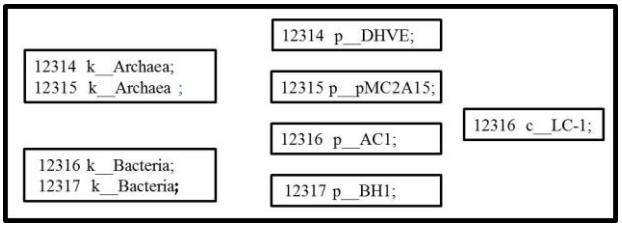

Fig. 13 Seven clusters of the microorganisms based on their taxonomic level.

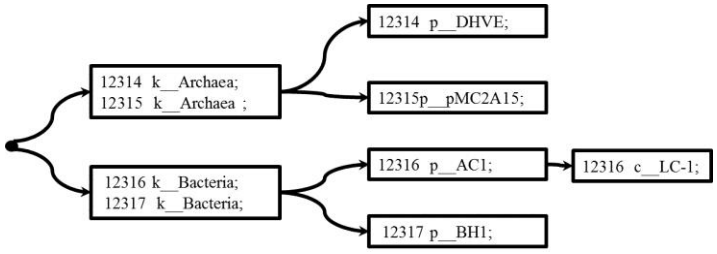

Fig. 14 A tree structure of the database.

Chapter 4

The Evaluation Framework

The taxonomic assignment framework, presented in (Nicholas A. Bokulich, 2014) and which is based on the IPython Notebook (Ragan-Kelley, et al., 2013), is used to evaluate the quality of a taxonomic classification. It does this by comparing the observing taxonomic assignments with other taxonomic classification methods (BLAST, UCLUST, RDP, MOTHUR, and RTAX). In my study, the evaluation framework is used to evaluate the HMM method and compare with other current taxonomic classification methods. The evaluation framework takes as input a user generated taxonomy file (for instance one generated by the HMM program) and a BIOM table containing the known taxonomies of data passed to the HMM program. The output of this step is another BIOM table that contains the merged data. Then, the evaluation framework compares the user generated data with the know taxonomy data, and then compares these annalistic results with the other methods.

The accuracy of a taxonomy-classification method can be measured in terms of precision, recall, F-measure, β diversity correlation and β diversity (the dissimilarity among samples). Precision, recall, and F-measure are qualitative measures, which are used to identify whether a microorganism is present or not. Precision, which is known as specificity, is the fraction of taxonomies retrieved from the database that are correctly identified. Recall, which is also known as sensitivity, is the fraction of taxonomies retrieved from the database that are identified. F-measure is the harmonic mean of precision and recall. Each of these measures is defined below:

Precision = true positive / (true positive + false positive)

Recall = true positive / (true positive + false negative)

F-measure = 2 × Precision × Recall / (Precision + Recall)

A true positive represents a taxonomy retrieved from the database that exists in the communities of microorganisms. A false positive represents a taxonomy retrieved from the database that does not exist in the microbial communities. A false negative represents a taxonomy that does not exist in the database, nor does it exist in the microbial communities. Usually, precision and recall cannot be discussed in isolation. If the value of recall increases, the corresponding value of precision will decrease. This indicates the increase in the search range of similar patterns in a database, which tolerates lower limitations (higher E-values), however, this will result in a lower precision. On the contrary, lower recall will result in a higher precision value.

β diversity correlation and β diversity are quantitative measures. β diversity is used to identify the abundance of microorganisms in a microbial community. Correlation is an indicator that uses abundance to how similar a test group and a known group are. Since some factors, like primer bias, can affect the ability of a taxonomic classification method by assigning taxa on each taxonomic rank, it could result in poor abundances. β diversity is another measure of abundance, but it is mainly used to distinguish species among two samples in a natural community. Bray-Curtis dissimilarity (Bray & Curtis, 1957) is one of the most well-known ways to quantify the β diversity in a naturally occurring community. The formula for calculating the Bray-Curtis dissimilarity is defined as:

$$d_{ij} = \frac{\sum_{k=1}^{n} |x_{ik} - x_{jk}|}{\sum_{k=1}^{n} (x_{ik} + x_{jk})}$$

In this equation, the number of microorganisms of the species k in the sample i is denoted by x_{ik}. Likewise, the number of microorganisms of the species k in the sample j is denoted by x_{jk}. d_{ij} is the degree of dissimilarity between the sample i and the sample j. The result will range between 0 and 1. If all the species in the samples are identical, d_{ij} will be 0 and if all the species are distinct, d_{ij} will be 1. Different dissimilarities among samples will form a distance matrix. In order to obtain a quick understanding of disparity among these samples, Principle Coordinate Analysis (PCoA) can help to visualize the dissimilarities of multidimensional data present in the distance matrix (Krzanowski, 1988). A simple example of a distance matrix and its PCoA plot is shown in Fig. 15. In this matrix, S_i represents a sample i, which consists of unknown microorganisms. The value in the distance matrix between S_1 and S_2 indicates how dissimilar the species in these two samples are. Based on Fig. 15, the species between S_2 and S_3 are far more dissimilar than the species between S_1 and S_2 and the species between S_1 and S_3 are more similar than the species between S_2 and S_3. When using the distance matrix to see how dissimilar samples are, the PCoA plot, which is an intuitive method, can also be used to visualize the differences between samples. In the PCoA plot in Fig. 15, the distance between S_1 and S_3 is greater than the distance between S_1 and S_2. Therefore, the species between S_1 and S_3 are very different.

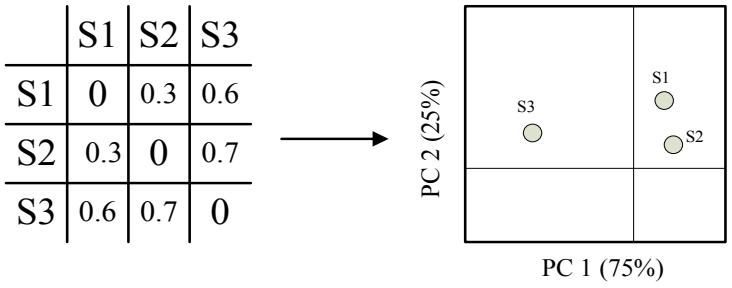

Fig. 15 An example of a distance matrix and its PCoA plot.

When two PCoA plots are generated from two different data sets, Procrustes analysis can be used to analyze whether the patterns on the of PCoA plots, which are based on the Bray-Curtis dissimilarity conclusions, are consistent. Procrustes analysis is a technique, which takes two different coordinate sets with corresponding points as input, and transforms one of the coordinate sets by rotating, scaling, and translating it to minimize the distance between the corresponding points of the two shapes. In order to determine how significant the divergence between each shape is, a goodness-of-fit (M^2) is used to summarize the discrepancy between these shapes. Usually, a smaller M^2 has a smaller discrepancy between the two shapes.

The test sequence data used in the calculations described in this framework were obtained from the GitHub repository: https://github.com/gregcaporaso/short-read-tax-assignment/. The sources of the test sequence data came from three different communities: a simulated community, a mock community, and a natural community.

For the simulated communities, "Sequences were extracted from type strains contained in the Ribosomal Database Project 16S rRNA database, Genbank, or where possible from the exact strain added to the mock communities" (Nicholas A. Bokulich, 2014). Each of the sequences had all defects removed that were caused by sequencing errors and PCR bias errors. This allowed for the most accurate assessment of a taxonomic classification by a classifier.

The mock communities were constructed from a group of sequences, which were extracted from bacteria and fungi. The sequences were generated by an Illumina or 454 sequencing machine. In this community, taxonomies of microbial sequences we have already known.

The sequences in the natural community were bacterial 16S rRNA or fungal ITS samples from a range of biological sites (wine, beer, cheese, and soils). Since the composition of the mock communities was so simple, the natural communities were built to reflect the complexity of the real world of microbial communities. In addition, taxonomies of microbial sequences in the natural community are not assigned by biologists. For the purpose of this study, only the sequences in the mock communities and one natural community were analyzed.

Chapter 5

Experiment Results

The evaluation results for the Hidden Markov Models (HMMs) method are presented here with different E-values and different classifiers. E-value is the only parameter that would affect the accuracy of the taxonomic classification in the HMMs method. In order to observe how the E-value influenced the accuracy of taxonomy classification, the range of E-values was adjusted from 100 to 1e-30. Since there are only slight differences in the accuracy of the results for qualitative and quantitative measures with the E-values between 100 to 1 and 1 to 1e-30, only three E-values are shown below with values of 100, 1, and1e-30.

Table 1 shows the precision, recall, F-measure, and correlation for eight samples of mock communities. The E-value used to generate the results was set to 100. The values for recall range from 0.1052 to 0.7895 and the average value of it is 0.5478. Since the recall value from the Broad-1 sample was lower than the other samples, it was excluded from the calculation of the average recall value, which caused the average value to go up from 0.5478 to 0.6076. The results indicate that the percentage of unknown microbes assigned taxonomies increased from approximately 55% to 61%, which means more unknown microbial genes could still be discovered.

The values for precision range from 0.0718 to 0.2857 and the average value equals 0.1579. This average value shows that the number of microbes that were correctly identified only account for 15.8%, which is significantly lower than the other methods, see Fig. 16. To see the overall qualitative analysis, the range of the F-measure values shown in Table 1 is from 0.1271 to 0.3488. Compared to other methods, this range is also

significantly lower, see Fig. 18. For the correlation value in the last column, the lowest value is -0.4162 and the highest value is 0.3455. The average value of correlation is 0.0995. This indicates that the observed abundances that are close to the referenced abundances is 9.95%. When the Broad-1 is excluded, the average value rises to 0.1731. By excluding the Broad-1 sample, the correlation value improved.

In order to easily read the tables, the extreme values in each column are bolded. To understand if the proposed classifier is good or not, these results are compared with other pre-existing methods using box and whisker plots. Box and whisker plots are used to show the average values of different classifiers and the gaps between each average value. Fig. 16, Fig. 17, Fig. 18, and Fig. 19 represent the box and whisker plots of various classifiers with precision, recall, F-measure, and correlation. Here the HMM method is called hmmtax. In Fig. 16, 17, and 18, the red line in each box represents an average value, the top of each box represents 75% of the data that reached that value, and the bottom of each box represents 25% of the data that reached that value. The average precision value, the average recall value, and the average F-measure value of the hmmtax are all lower than the other methods used, like the RTX, the BLAST, the RDP, and the MOTHUR. However, the average correlation value in Fig. 19 is closer to the other methods. Therefore, although the qualitative analysis of accuracy of taxonomic classification for the hmmtax classifier does not perform as well as the other classifiers, the abundances in a sample can still be distinguished to a certain degree by the hmmtax classifier.

Table 1 8 samples with precision, recall, F-measure, and correlation results when E-value is 100.

	Precision	Recall	F-measure	Correlation
Broad-1	**0.2857**	**0.1052**	0.1538	**-0.4162**
Broad-2	0.1548	0.6842	0.2524	0.1169
Broad-3	0.2238	**0.7895**	**0.3488**	**0.3455**
S16S-1	0.1373	0.6087	0.224	0.1853
S16S-2	0.1899	0.6522	0.2941	0.1396
Turnbaugh-1	**0.0718**	0.5556	**0.1271**	0.1887
Turnbaugh-2	0.0889	0.4444	0.1481	0.1105
Turnbaugh-3	0.1111	0.5185	0.1830	0.1257
Average	0.1579	0.5478	0.2164	0.0995

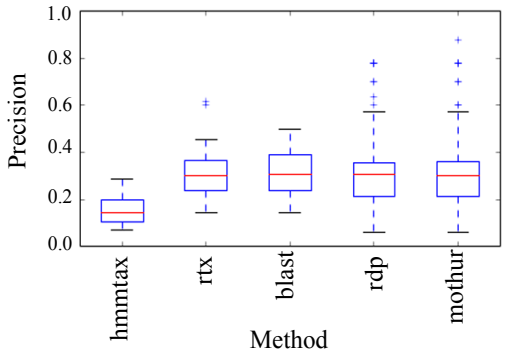

Fig. 16 The precision results with different classifiers when the E-value of the hmmtax is 100.

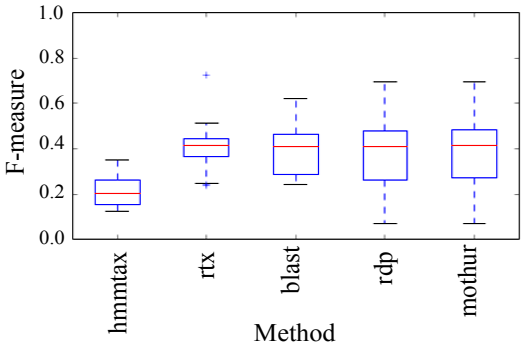

Fig. 18 The F-measure results with different classifiers when the E-value of the hmmtax is 100.

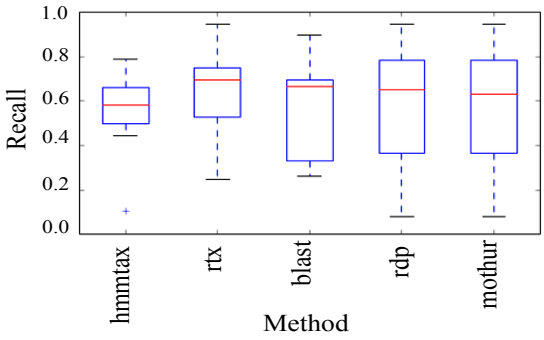

Fig. 17 The recall results with different classifiers when the E-value of the hmmtax is 100.

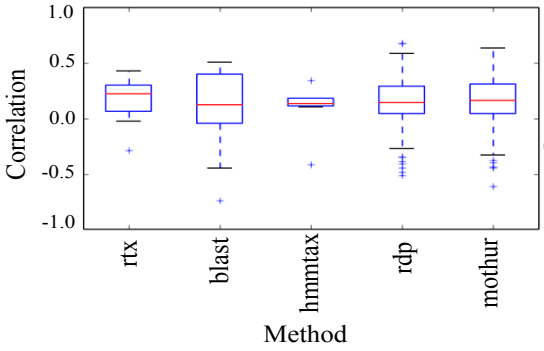

Fig. 19 The correlation results with different classifiers when the E-value of the hmmtax is 100.

For the next set of data shown in Table 2, the E-value was adjusted to 1 for precision, recall, F-measure, and correlation. The range of the precision values is from

0.0718 to 0.2857 and the average precision value is 0.1579. The range of the recall values is from 0.1052 to 0.7895 and the average recall value is 0.1052. The range of the F-measure values is from 0.1271 to 0.3488 and the average F-measure value is 0.2164. The range of the correlation values is from -0.4162 to 0.3455 and the average correlation value is 0.0995. Fig. 20, Fig 21, Fig 22, and Fig. 23 represent the box and whisker plots of various classifiers with precision, recall, F-measure, and correlation. When comparing these results with those in Table 1, the accuracy of taxonomic classification did not change.

Table 2 The precision, recall, F-measure, and correlation results along with eight samples when E-value = 1.

	Precision	**Recall**	**F-measure**	**Correlation**
Broad-1	**0.2857**	**0.1052**	0.1538	**-0.4162**
Broad-2	0.1548	0.6842	0.2524	0.1169
Broad-3	0.2238	**0.7895**	**0.3488**	**0.3455**
S16S-1	0.1373	0.6087	0.2240	0.1853
S16S-2	0.1899	0.6522	0.2941	0.1396
Turnbaugh-1	**0.0718**	0.5556	**0.1271**	0.1887
Turnbaugh-2	0.0889	0.4444	0.1481	0.1105
Turnbaugh-3	0.1111	0.5185	0.1830	0.1257
Average	0.1579	0.1052	0.2164	0.0995

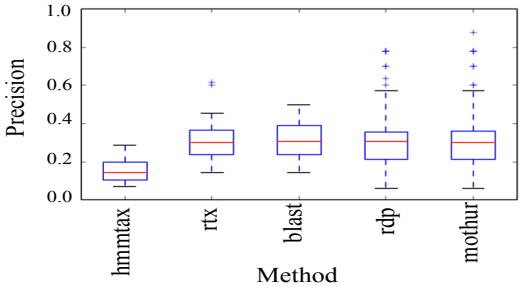

Fig. 20 The precision results with different classifiers when the E-value of the hmmtax is 1.

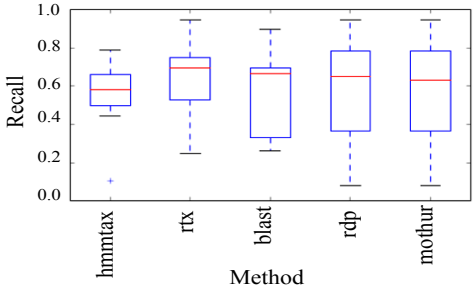

Fig. 21 The recall results with different classifiers when the E-value of the hmmtax is 1.

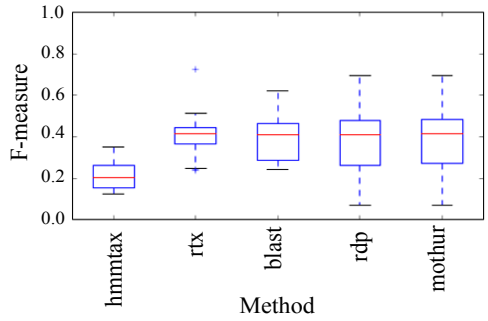

Fig. 22 The F-measure results with different classifiers when the E-value of the hmmtax is 1.

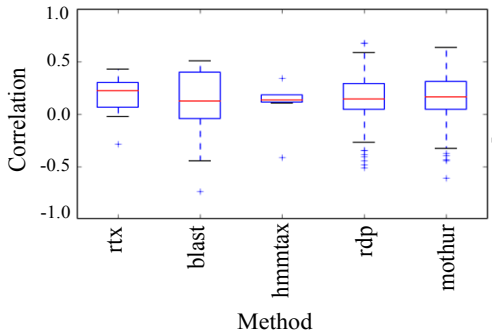

Fig. 23 The correlation results with different classifiers when the E-value of the hmmtax is 1

In the last of data, the E-value was adjusted to 1e-30 for precision, recall, F-measure, and correlation, which are shown in Table 3. The range of the precision values is from -1 to 0.2857 and the average precision value is -0.2598. The range of the recall values is from -1 to 0.7895 and the average recall value is -0.1019. The range of the F-measure values is from -1 to 0.3488 and the average F-measure value is -0.2409. The value of -1 represents microbial sequences where no taxonomies are assigned. The range of the correlation values is from -0.4162 to 0.3455 and the average correlation value is -0.16. Fig. 24, Fig. 25, Fig. 26, and Fig. 27 represent the box and whisker plots of various classifiers with precision, recall, and F-measure. When comparing these results with those in Table 2, the accuracy of taxonomic classification decreases.

Table 3 The precision, recall, F-measure, and correlation results along with 8 samples when the E-value = 1e-30

	Precision	**Recall**	**F-measure**	**Correlation**
Broad-1	**0.2857**	0.1052	0.1538	-0.4163
Broad-2	0.1406	0.4739	0.2169	-0.441
Broad-3	0.2238	**0.7895**	**0.3488**	**0.3430**
S16S-1	-1	-1	-1	-0.1865
S16S-2	0.2	0.2609	0.2264	-0.1157

Turnbaugh-1	0.0718	0.5556	0.1271	-0.1545
Turnbaugh-2	-1	-1	-1	-0.1545
Turnbaugh-3	-1	-1	-1	-0.1545
Average	-0.2598	-0.1019	-0.2409	-0.16

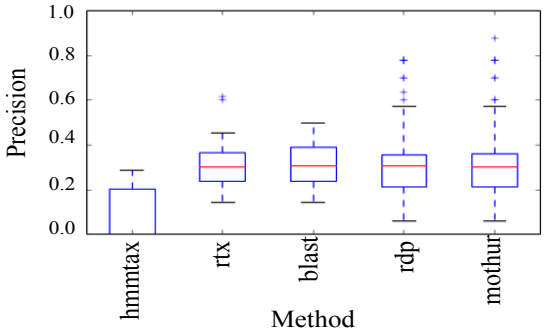

Fig. 24 The precision results with different classifiers when the E-value of the hmmtax is 1e-30.

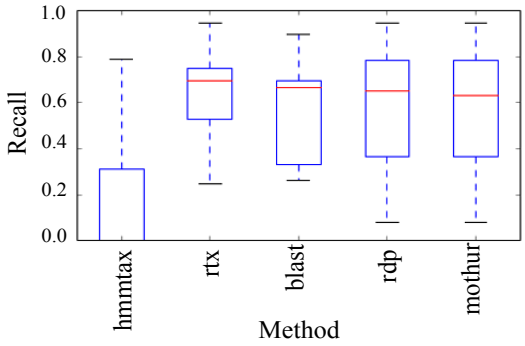

Fig. 25 The recall results with different classifiers when the E-value of the hmmtax is 1e-30.

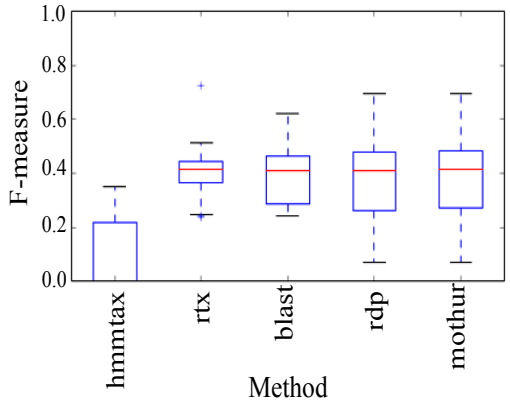

Fig. 26 The F-measure results with different classifiers when the E-value of the hmmtax is 1e-30.

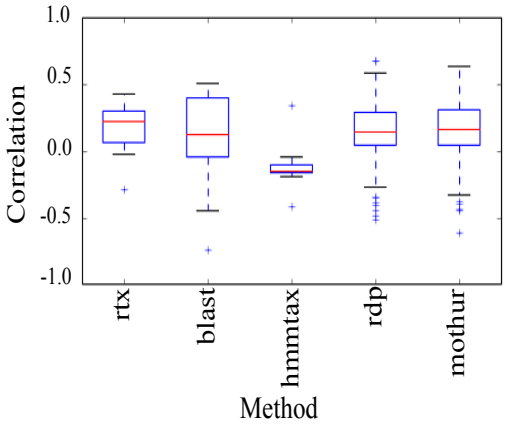

Fig. 27 The correlation results with different classifiers when the E-value of the hmmtax is 1e-30.

Although the hmmtax did not have better results on the qualitative analysis, it still has the ability to detect the abundances in a natural community. Table 4 shows the $M^2$of the Study_449 from the natural community with each classifier and its parameter. As you can see, the M^2value for the hmmtax where the E-value=1e-30, is higher than the other two E-values of 100 and 1. Also, it is better than the other classifiers, like the RDP with

parameters 0.1 and 0.5 and the MOTHUR with parameters 0.1 and 0.5. However, the best result for M^2 occured when the E-value was adjusted to 1e-30.

When the E-value keeps going down to 1e-40 (not shown), the HMMs method cannot find any taxonomy in the reorganized Greengenes database. Hence, the M^2 value would be larger than any M^2 value listed in the table. The reason behind this could be that the sequences in the Greengenes database are at 97% identity. If the HMMs method is applied to a database, which consists of sequences at 99% identity, the taxonomic coverage will be broader. Then, the M^2 value will increase with a lower E-value utilized in the HMMs method.

Table 4 The M^2 with methods and parameters.

Data set	$\boldsymbol{M^2}$	**Method**	**Parameters**
Study_449	0.047	rtax	single
Study_449	0.104	blast	1.0
Study_449	0.120	rdp	1.0
Study_449	0.134	Mothur	1.0
Study_449	0.195	rdp	0.8
Study_449	0.196	Mothur	0.8
Study_449	**0.211**	**hmmtax**	**1e-30**
Study_449	0.234	rdp	0.1
Study_449	0.235	mothur	0.1
Study_449	0.237	rdp	0.5
Study_449	**0.240**	**hmmtax**	**100**
Study_449	**0.240**	**hmmtax**	**1**
Study_449	0.250	mothur	0.5

Chapter 6

Conclusions

Hmmtax classifier is a taxonomic classification for microbial gene sequences. It searches short 16S rRNA sequence reads of unknown origin against the tree structure of the greengenes database, applying Hidden Markov Models at each internal node in the tree to determine which potential branch from a given point is most likely to represent the taxonomic origin of the sequence. To evaluate the accuracy of the taxonomic classification obtained from hmmtax, the metrics applied in this study included qualitative analysis, based on precision, recall, F-measure, and quantitative analysis, which consists of correlations and β diversity. I compared hmmtax to other pre-existing classifiers, including BLAST, RDP, RTAX, MOTHUR, and UCLUST, to see if the hmmtax classifier could achieve better taxonomic assignments than these methods.

On most metrics, the hmmtax results are not as good as the other methods. These results are shown in Tables 1, 2, and 3. The highest precision value achieved was 0.2857 and the highest recall achieved was 0.7895. These values are lower than the other methods, suggesting that hmmtax is not a useful improvement over other approaches.

Hidden Markov Models have been primarily used to model protein sequences, which composed of 20 amino acids. In nucleotide sequences, there are on four possible choices: A, U/T, C, and G. In addition, the profile Hidden Markov Model regards each nucleotide as an independent nucleotide in the sequence, which means that there are no correlations in the sequence, but the 16S rRNA sequences contain self-complementary base-pairs that form the secondary structure of the 16S rRNA molecule. I suspect that these two points might be detrimental for taxonomic assignments of rRNA sequences

with Hidden Markov Models because there are not a lot of states (four) and it is possible for sequences to have higher similarity, even if they are dissimilar to each other in secondary structures, and sequence is probably more important than structure when determining the relatedness of a pair of sequences.

In the future, one possible approach that I might try to improve the accuracy of taxonomic classification with Hidden Markov Models is the use of the infernal software package for making assignments. This tool is specific to 16S rRNA, which is why I didn't start with it (I wanted a general taxonomic classifier) but not only considers the primary nucleotide sequence, but also considers the secondary structure of 16S rRNA sequence. I think that incorporating this additional information is likely to improve taxonomic assignment beyond what is achieved with a standard Hidden Markov Model.

Although hmmtax did not perform well in the qualitative analyses, it did achieve good performance in correlation analyses when E-value is 100 and 1 and β diversity analyses. When comparing the value of β diversity of hmmtax with other methods, it is higher than some of classifiers, meaning that it is better at determining the abundance profile of taxa in a sample that the other methods. I suspect that this is because the profile Hidden Markov Models were built from many sequences at each taxonomic level (a multiple sequence alignment of all of the 16S belonging to a given taxon), so they are better able to recognize diverse sequences that group into the same taxa than the other approaches. This translates into better estimations of taxa abundance. Hmmtax in its current form may therefore be a useful complement to the other taxonomic assignment methods here, and I suspect that future additions (as discussed in the previous paragraph) may increase its performance on other benchmarks.

References

Allberts, B., Bray, D., Lewis, J., Raff, M., Boberts, K., & Watson, J. D. (2002, September). *Molecular Biology of the Cell* (4th ed.). New York: Garland Science.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic Local Alignment Search Tool. *J. Mol. Biol., 215*, pp. 403-410.

Biello, D. (2010). Meet the Microbes Eating the Gulf Oil Spill . *Scientific American*.

Bokulich, N. A., Rideout, J. R., Patnode, K., Ellet, Z., McDonald, D., Wolfe, B., . . . Caporaso, J. G. (2014). An extensible framework for optimizing classification enhances short-amplicon taxonomic assignments. *Nature, underreview*.

Bray, J. R., & Curtis, J. T. (1957, October). An Ordination of the Upland Forest Communities of Southern Wisconsin. *Ecological Monographs, 27*(4), pp. 325-349.

Caporaso, J. G., Kuczynski, J., Stombaugh, J., Bittinger, K., Bushman, F. D., Costello, E. K., . . . Knight, R. (n.d.). *Greengenes 13_5*. Retrieved from QIIME: http://qiime.wordpress.com/2013/05/20/greengenes-13_5/

Clarridge, J. E. (2004). Impact of 16S rRNA Gene Sequence Analysis for Identification of Bacteria on Clinical Microbiology and Infectious Diseases. *Clinical Microbiology Reviews, 17*(4), pp. 840-862.

Department of Health and Human Services. (2009). *Understanding Microbes in Sickness and in Health.* Maryland: National Institutes of Health.

DeSantis, T. Z., Hugenholtz, P., Larsen, N., Rojas, M., Brodie, E. L., Keller, K., . . . Andersen, G. L. (2006). Greengenes, a Chimera-Checked 16S rRNA Gene Database and Workbench Compatible with ARB. *72*(7), pp. 5069–5072.

Durbin, R., Eddy, S., Krogh, A., & Mitchison, G. (1998). *Biological Sequence Analysis.* Cambridege University Press.

Eddy, S. R., & Wheeler, T. J. (2013, May). Retrieved from HMMER: http://hmmer.org

Edgar, R. C. (2010). Search and clustering orders of magnitude faster than BLAST. *Bioinformatics, 26*(19).

Harley, S. (2009). *Antelope Island Field Trip: Life in the Great Salt Lake.* Retrieved from Weber State University Department of Botany: http://faculty.weber.edu/sharley/AIFT/GSL-Life.htm

Koonin, E. V., & Galperin, M. Y. (2003). Chapter 2, Evolutionary Concept in Genetics and Genomics. In *Evolution - Function: Computational Approaches in Comparative Genomics.* Boston: Kluwer Academic. Retrieved from http://www.ncbi.nlm.nih.gov/books/NBK20255/

Krzanowski, W. J. (1988). *Principles of Multivariate Analysis. A User's perspective.* Oxford: Oxford Univ Press.

Liu, A., DeSantis, T. Z., Andersen, G. L., & Knight, R. (2008, August). Accurate taxonomy assignments from 16S rRNA sequences produced by highly parallel pyrosequencers. *Nucleic Acids Research, 36*(18).

Liu, L., Li, Y., Li, S., Hu, L., He, Y., Pong, R., . . . Law, M. (2012, April). Comparison of Next-Generation Sequencing Systems. *Journal of Biomedicine and Biotechnology, 2012*.

Needleman, S. B., & Wunsch, C. D. (1970, July). General Method Applicable to the Search for Similarities. *J. Mol. Bwl., 48*, pp. 443-453.

Nicholas A. Bokulich, J. R. (2014). An extensible framework for optimizing classification enhances short-amplicon taxonomic assignments. *Nature Methods, Under review*.

Pogacic, T., Kelava, N., Zamberlin, S., Dolencic-Spehar, I., & Samarzija, D. (2010). Methods for Culture-Independent Identification of Lactic Acid Bacteria in Dairy Products. *Food Technol. Biotechnol, 48*(1), pp. 3-10.

Rabiner, B. H., & Juang, L. R. (1986, Junuary). An Introduction to Hidden Markov Models. *IEEE ASSP Magazine, 3*(1), pp. 4-16.

Ragan-Kelley, B., Walters, W. A., McDonald, D., Riley, J., Granger, B. E., Gonzalez, A., . . . Caporaso, J. G. (2013, May). Collaborative cloud-enabled tools allow rapid. *ISME J, 7*, pp. 461-464.

Schloss, P. D., Westcot, S. L., Ryabin, T., Hall, J. R., Hartmann, M., Hollister, E. B., . . . Weber, C. F. (2009, Dec.). Introducing mothur: Open-Source, Platform-Independent, Community-Supported Software for Describing and Comparing Microbial Communities. *APPLIED AND ENVIRONMENTAL MICROBIOLOGY, 75*(23), pp. 7537-7541.

Sciences, U. o. (2014). *Microbes At Work*. Retrieved from Genetic Science Learning Center: http://learn.genetics.utah.edu/content/gsl/microbes/

Smith, T. F., & Waterman, M. S. (1981). Identification of common molecular subsequences. *J. Mol. Biol., 147*, pp. 195-197.

Soergel, D. A., Dey, N., Knight, R., & Brenner, S. E. (2012). Selection of primers for optimal taxonomic classification of environmental 16S rRNA gene sequences. *The ISME Journal, 6*, pp. 1440-1444.

Talaro, K. P. (1999). *Foundations in Microbiology.* McGraw-Hill Education.

Wang, Q., Garrity, G. M., Tiedje, J. M., & Cole, J. R. (2007, Aug.). Naïve Bayesian Classifier for Rapid Assignment of rRNA Sequences into the New Bacterial Taxonomy. *Appl. Environ. Microbiol, 73*(16), pp. 5261-5267.

Ward, D. M., Weller, R., & Bateson, M. M. (1990, May). 16S rRNA sequences reveal numerous uncultured microorganisms in a natural community. *Nature, 345*, 63-65.

Wijffels, R. H., & Barbosa, M. J. (2010). An Outlook on Microalgal Biofuels. *Science, 329*, 796.

Wu, P. C., Lau, S. K., Teng, J. L., Tse, H., & Yuen, K. (2008). Then and now: use of 16S rDNA gene sequencing for bacterial identification and discovery of novel bacteria in clinical microbiology laboratories. *Clinical Microbiology and Infection, 14*(10), pp. 908-934.

Zvelebil, M., & Baum, J. O. (2008). *Understanding Bioinformatics.* New York: Garland Science, Taylor & Francis Group, LLC, an informa business.