Multiple Instance Learning of Clinical Phenotype Based on Metagenome-Wide Association Study

Nathan LaPierre
Department of Computer Science
George Mason University
Fairfax, Virginia 22030
Email: nlapier2@masonlive.gmu.edu

Mohammad Rahman
Department of Computer Science
George Mason University
Fairfax, Virginia 22030
Email: arif.at.iut@gmail.com

Huzefa Rangwala
Department of Computer Science
George Mason University
Fairfax, Virginia 22030
Email: rangwala@cs.gmu.edu

***Abstract*—**

We demonstrate a computational method to predict the clinical phenotypes of a patient from raw metagenomic sequence read data. We compared two state of the art programs for annotating the sequence data, UCLUST and Kraken, and using their output for feature generation. We apply these programs to a set of over 1.3 million reads from 904 patients, some of whom have liver cirrhosis, encephalopathy due to liver cirrhosis, or neither disease. Once the reads have been processed by UCLUST or Kraken, we use Support Vector Machines to setup the clinical phenotype prediction problem. We find that too many false negatives are being predicted by the classifier. In order to address the issue, we scale features to improve the classification model and evaluate the end results on a held-out test set. We demonstrate our approach works quickly and accurately with an 85.64% success rate when we use the UCLUST representation. We also find that UCLUST generally performs better than Kraken, with the latter having an 80.66% success rate. We also test our classifier on several subsets of the data, with success rates ranging from 69.81% to 96.72%.

I. Introduction

The human body is one of the most densely populated microbial environments in the world, with human host cells interacting with more than 10^{14} microbial cells. Collectively, the human and microbial cells are referred to as the *human microbiome* [1], [2]. Biotechnological advances can interrogate the microbial communities present across three primary dimensions: (i) clinically different patients like those suffering from liver disease versus healthy controls (ii) different patient sites and tissues like skin and the gastrointestinal tract and (iii) across different patient visits [3], [4] and before/after treatment. Methods to query this microbial data include DNA sequencing of specific marker genes such as the 16S rRNA marker genes [5], sequencing the entire pool of microbial genomes at once (metagenomics), or sequencing the mixture of expressed gene transcripts (metatranscriptomes). These sequencing methods along with proteomics and mass spectrometry allow us to study the biotransformations caused by the microbial communities. Several researchers and clinicians have embarked on studies of the pathogenic and clinical role played by the microbiome with respect to human health and disease conditions.

In this study, we specifically developed a computational method to predict the clinical phenotype of a patient i.e., a pipeline to predict whether or not the patient has a specific type and level of a disease. This pipeline is able to make the phenotypic predictions from input metagenomic sequences using post-processing with unsupervised binning approaches (clustering) or supervised search based taxa profiling approaches. This method is designed to run quickly and produce accurate and sensitive results. We compare and evaluate two state-of-the-art approaches, one for binning (UCLUST [6]) and one for taxonomic profiling (KRAKEN [7]) to represent the input metagenome sequences as features for classification. The classification formulation uses a binary support vector machine based classifier [8]. We also evaluate the performance of preliminary feature scaling and engineering for classification purposes.

We specifically evaluated our approach on microbiome samples of patients suffering from hepatic encephalopathy due to liver cirrhosis. We found that UCLUST performed better than Kraken on our data set, although Kraken was faster. We were also able to engineer the data to improve the accuracy of the classifier. Our pipeline had 85.64% accuracy using UCLUST and 80.66% accuracy using Kraken.

II. Background

A. Human Microbiome and Metagenomics

The combination of human host cells and microbial cells that govern several facets of human health and pathology are referred to as collectively by the term *human microbiome* [9]. Using today's sequencing technologies we have the capacity to determine the DNA sequences of these co-existing microbial communities. However, current genomic technologies do not provide the complete genome for each individual microbe, but short, contiguous subsequences from random positions of the genome. These short subsequences are referred by "reads". Metagenome sequence assembly is defined as the process of taking the different sequence reads from the various microbial organisms to produce long contiguous sequence of DNA for each individual organism within the community mixture. The assembly process involves identifying overlapping parts of different reads to order them and separate them in organism-specific larger sequences.

The metagenome assembly problem is known to be challenging due to the similarity of genomes from the different microbes, differing abundance, diversity, complexity and varying genome lengths of never-sequenced before microbes within the different microbiome samples. DNA sequencing machines are very high throughput and produce Terabytes of data per run and also produce reads with poor quality [9]. Several studies have highlighted the challenges associated with the metagenome analysis and assembly problem by performing a simulation study to determine the feasibility of solving the metagenome assembly problem [10].

Targeted metagenomics or 16S rRNA gene sequencing provides a first step for the quick and accurate characterization of microbial communities. 16S sequences are marker genes, which exists in most microbial genomes and have a conserved portion for detection (primer development) and a variable portion that allows for categorization within different taxonomic groups [11]. Targeted metagenomics are also effective in detecting species with low abundances. However, they may not be good at discovering unique species (orphans) that have never been sequenced before.

B. Microbiome Informatics

Since the human microbiome project [1] and release of publicly available metagenomic datasets began, a host of methods have been developed for the analysis of assembled metagenomes and input sequence reads (before assembly) obtained from 16S rRNA genes and whole metagenomes. The relationship between microbiome and human health can be characterized by first identifying the content, abundance, and functionality of the microbes within the samples. Several computational approaches (surveyed here [12]) have been developed for handling the vast amount of metagenomic data to solve two related problems: (i) clustering or binning, and (ii) taxonomy profiling methods.

1) Binning Methods: The "binning" problem involves grouping input short reads such that reads within a group are similar to each other. This process may lead to groups that are organism-specific and is unsupervised in nature, being referred by clustering in the data mining community. The binning process does not attempt to provide an automated labeling of the input reads. The clusters/bins/groups obtained from an input metagenome sample is referred to as Operational Taxonomic Units (OTUs), and the number of OTUs gives an approximation of species diversity in a sample [13], [14], [15]. These approaches are not constrained due to the absence of a complete coverage in taxonomic databases. Some environmental samples contain microbial organisms that have never been cultured in a laboratory, and thus those organisms do not exist in genomic databases. As such, binning of sequence reads has several advantages: (i) it can lead to an improved metagenome assembly, (ii) it can be used for computing species diversity metrics [16] and (iii) the reduced computational complexity within several work-flows that analyze only cluster representatives, instead of individual sequences within a sample.

CD-HIT [17], UCLUST [6], CROP [18], MC-MinH [19] and MC-LSH [16] are some of the popular metagenome/sequence clustering approaches used for binning. UCLUST, MC-MinH and MC-LSH are greedy approaches that achieve computational efficiency by using either hash-based indexing or matching of gap-less sequences called seeds (instead of expensive sequence alignment) and followup with an incremental clustering approach that does not involve comparing all pairs of input sequences. In Section III-B we discuss in detail the UCLUST [6] sequence clustering approach. We use UCLUST within our study, known to be a state-of-the-art metagenome clustering approach in terms of computational performance and accuracy of results.

2) Taxonomy Profiling Methods: The taxonomy profiling problem involves assigning a specific label (i.e., a phylogenetic group label) to sequence reads or assembled metagenome contigs [20]. A traditional approach for taxonomy profiling is to formulate a classification problem (supervised) that uses marker genes for identification of source organism of a sequence read or fragment [5]. Marker genes are highly conserved and provide accurate identification of the taxonomy class [21]. These approaches rely on a previously annotated reference dataset like RDP [22], [23], [24] or GreenGenes [25]. These methods provide valuable community estimates but are limited to specific reads or contigs (marker genes constitute a small fraction of a metagenomic sequence set) and have low sensitivity due to reliance on an incomplete and taxon-biased reference genome database.The RDP database implements a naive Bayes classifier that uses DNA-composition features to classify 16S RNA sequence reads into taxonomic classes as defined by Bergey's taxonomy [21].

Several comparative methods have been developed for the assignment of phylogenetic class based on principles of homology. Such methods align reads or contigs using BLAST [26], and assign taxonomy based on the best match with a reference database [27]. MEGAN [28] and MARTA [29] are metagenomic analysis and visualization programs that make the assignment based on multiple BLAST hits and optimized parameters. The MG-RAST [30] web server provides taxonomic and functional annotation by comparative searches performed across multiple reference databases. GAAS [31] is a novel BLAST-based tool that includes genome length normalization along with a similarity weighting for multiple BLAST hits to provide improved estimates.

Composition-based methods have also been developed. They extract key sequence features such as GC composition and k-mer frequencies and build supervised classification models using those features. PhyloPythia [32] uses a support vector machine framework [8] to classify long reads into taxonomic groups using a k-mer based kernel function. TETRA [33] correlates the k-mer pattern feature to different taxonomic groups. Kraken, which is used in this paper, searches a taxonomic database for the lowest common ancestor (LCA) of the genomes that contain k-mers from a sequence read. Phymm [20] trains an interpolated Markov model to characterize variable length subsequences specific to different taxonomic subgroups. In combination with BLAST, Phymm shows improved classification accuracy for short reads of 100 base pair (bp) length. Such Markovian models have been very successful in gene finding algorithms like Glimmer [34].

III. Methods

A. Overview

Figure 1 provides an overview of the developed computational pipeline. We obtain 16S metagenomic sequence reads from patients who had encephalopathy and liver cirrhosis, no encephalopathy but liver cirrhosis, and controls (with neither

[1] http://hmpdacc.org/

phenotype). The input sequence reads are annotated using a taxonomic profiling approach (Kraken) or an unsupervised binning approach (UCLUST). The output of this annotation phase features are generated, which are passed on the binary support vector machine classifier (SVMs). We used the svm-light implementation [35] to perform the classification of whether or not a patient had a clinical phenotype. We compared the classification performance of the SVM classifier with the two intermediate sequence representations obtained from UCLUST and Kraken. We also applied feature scaling and selection techniques to improve the classification results.

B. Unsupervised Clustering using UCLUST

In this work we evaluate the use of an unsupervised approach for representing our input metagenome sequence reads. Specifically, we use UCLUST [36] due to its superior performance in terms of run time and accuracy. UCLUST [36] follows a greedy, iterative clustering approach. As a first step, this approach identifies exact matches of fixed length between sequence pairs known as seeds. These seeds are then extended by allowing for a few mismatches and/or gaps between the aligned pairs. Seeds scoring above a certain threshold are chosen as high segment pairs (HSPs) and used for further processing. As such, the step eliminates a lot of pairwise comparisons.

Then UCLUST follows an incremental approach for assigning the input sequences to the different bins. The clustering solution is initialized as an empty list. Sequences are then incrementally added to the clusters existing within the list. Each unassigned input sequence is compared to the cluster representatives within the list using the fast indexing search technique that uses the HSPs. If a match is found with one of the cluster representatives, then that sequence is assigned to that particular cluster or the input sequence forms a new cluster. UCLUST ensures that the cluster representatives are sequences with the largest length.

C. Supervised Taxonomic Identification using Kraken

We also evaluate the use of a search based approach to taxonomically label our sequence reads. We use Kraken for this, due to its strong performance in terms of run time and precision. Kraken finds k-mers in an input sequence read and queries a database of OTUs. It then associates the read with the lowest common ancestor (LCA) of the genomes that contains k-mers from the read [37]. In order to reduce the number of false positives, Kraken does not classify reads for which it does not find sufficient evidence of any match.

This approach is designed to perform quickly and with high precision. However, it also involves downloading and building a large database to query. It is often infeasible to download and build the database in its entirety, so a smaller custom subset of the database is available for testing purposes. We used this subset of the entire database in this study (referred to as kraken-light).

D. Classification using Support Vector Machines (SVMs)

Support Vector Machines (SVMs) [8] are one of the most powerful and versatile binary classifiers used in myriad applications. For a binary classification problem, the support vector machine (SVM) [8] framework maximizes the separation between the two classes.

Given an input set of positive training instances and negative training instances, the SVM classifier learns a mathematical function f that maximizes the separation between the given positive and negative training instances. The function f will then be able to predict whether a new instance (considered a test instance) should have a positive or negative class label. For our formulation we use SVMs to make clinical phenotype prediction for each patient sample based on their input metagenome sequence samples represented with their clustering or taxa membership features.

For all the available microbiome samples, we have a set of reads processed via UCLUST or Kraken. Each patient is associated with a vector that is used by the SVM to help build f. Each row in the vector, called a feature, corresponds to a certain species, and the value for that row corresponds to how closely the read from the patient matched what the species was expected to look like. The exact process of assembling a feature vector for a patient depends on which data representation is being used. For UCLUST, we use each cluster seed as a feature. For each input sequence, we use the percentage match to its cluster seed as the value for that feature and average the sequence match if multiple reads from the metagenome samples are assigned to the same cluster by UCLUST. For Kraken, we use the taxonomic label associated with the read as the feature for the SVM. The percentage match of the read to that label is used as the feature value. Similar to UCLUST, we average the similarity score if multiple reads align to the same taxonomic class label. Our feature selection process involves discarding Kraken's unlabeled reads and UCLUST's clusters with only one read (singletons).

We used cross-validation and a held out set procedure (discussed in the next Section) to assess the performance of SVM classification using UCLUST and Kraken as intermediate representations.

IV. Materials

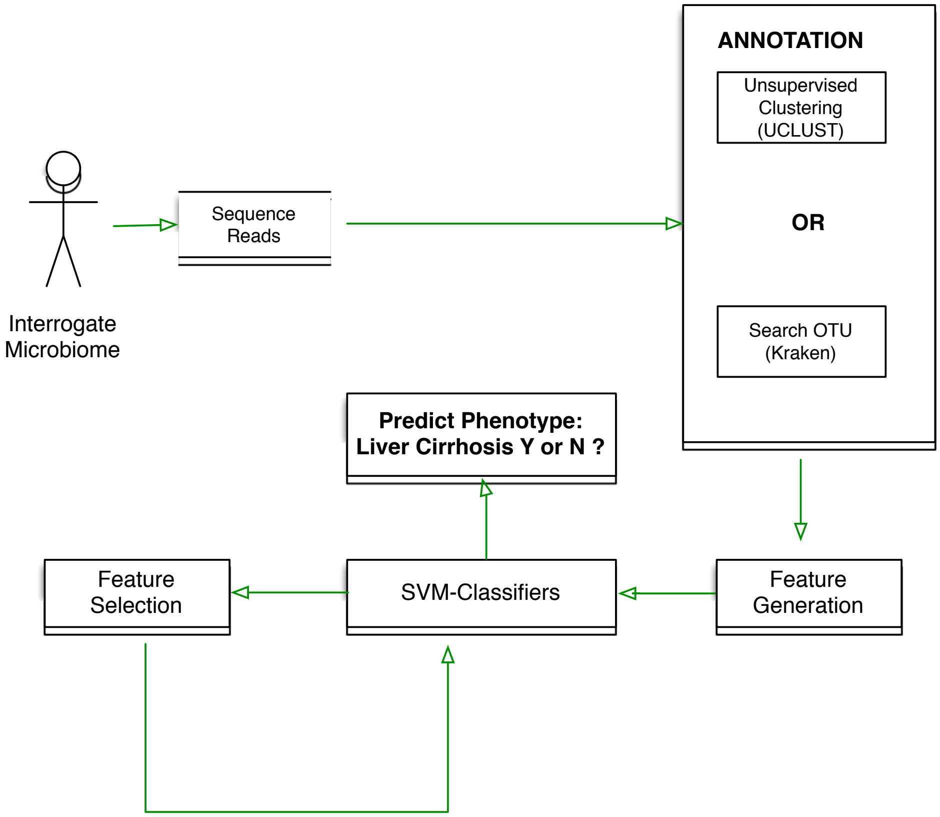
A. Dataset Description

We obtained 904 clinical microbiome samples from a study by Dr. Patrick Gillevet that relates to patients who suffered from hepatic encephalopathy due to liver cirrhosis. Specifically, the classification formulation was setup to distinguish between patients suffering with a specific clinical phenotype or not. There were a total of 239 patients with Encephalopathy due to liver cirrhosis (denoted by the "Encephalopathy" class), 590 patients with liver cirrhosis but no Encephalopathy (denoted by "No Encephalopathy") and 75 patients who were considered as control and did not have either of the clinical conditions (denoted by "Control"). 16S rRNA metagenomic sequence read data was obtained from patient stool samples. On average there were a total of 1,464 sequence reads of length 200-400 obtained per patient, with 1,323,016 total reads across all patients in our dataset.

B. Evaluation Metrics

We assess the performance of our classification pipeline in terms of correctness and execution time. Given the imbalanced

Fig. 1. Pipeline Overview.



nature of class distributions, the performance of binary classifiers was measured by F1 score, precision and recall, along with the standard accuracy metric. We discuss these standard metrics in brief.

Accuracy measures the percentage of instances that are classified correctly and can be represented by

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad (1)$$

where TP, TN, FP and FN represents true positives, true negatives, false positives and false negatives respectively.

Accuracy as an evaluation metric can be biased if one of the classes (positive or negative) has a larger number of examples than the other. Precision measures the percentage of positive predictions that were correct, whereas recall measures the percentage of positive examples that were correctly predicted (or retrieved). We can represent Precision and Recall by [38]:

$$Precision = TP/(TP + FP). \quad (2)$$

$$Recall = TP/(TP + FN). \quad (3)$$

The F1 score captures the trade offs between precision and recall in a single metric and is the harmonic mean of precision and recall [38], given by:

$$F1-Score = 2*(Precision*Recall)/(Precision+Recall). \quad (4)$$

C. Software and Hardware Details

We used the Argo computing cluster available at George Mason University. The representation generation phase using UCLUST and Kraken and and the classification phase were run on one of the compute nodes available on the cluster. The cluster is configured with 35 Dell C8220 Compute Nodes, each with dual Intel Xeon E5-2670 (2.60GHz) 8 core CPUs, with 64 GB RAM. (Total Cores 528 and 1056 total threads, RAM >2TB)[39].

Source codes for UCLUST [6] and KRAKEN [7] were downloaded from their respective websites[2] and compiled on the Argo platform. Kraken aligns reads to an OTU database. The standard Kraken database is 160GB in size. Due to computational limitations we used the custom, small sized 4GB database in this feasibility study [37].

For the SVM-based classification[35] we used the popular SVM-Light [35] source code, which is publicly available [3]. The linear kernel was used and the regularization parameters were set to their default values.

D. Experimental Protocol

For evaluating the performance of our binary phenotypic classifiers, we split the patient samples into a training set containing 80% of the patients and the test set containing 20%

[2] UCLUST: http://www.drive5.com/uclust/downloads1_2_22q.html Kraken: https://ccb.jhu.edu/software/kraken/

[3] http://svmlight.joachims.org/

of the patient samples. Every fifth sample was placed in the test set, so the relative class sizes were roughly consistent between the training and test sets. We also performed leave-one-out cross validation (LOOCV).

In the next section, we discuss the accuracy of our classifier with regards to predicting the clinical phenotype of a given patient using either the unsupervised clustering representation with UCLUST or supervised OTU representation with Kraken. We present results for classifiers distinguishing patients in the “Encephalopathy” class versus the other classes, effectively determining whether or not a patient had Encephalopathy regardless of whether they had liver cirrhosis. We also present pairwise one-versus-one classification results that compared two phenotypes within the dataset, i.e. Encephalopathy versus No Encephalopathy, Encephalopathy versus Control and No Encephalopathy versus Control.

V. Experimental Results

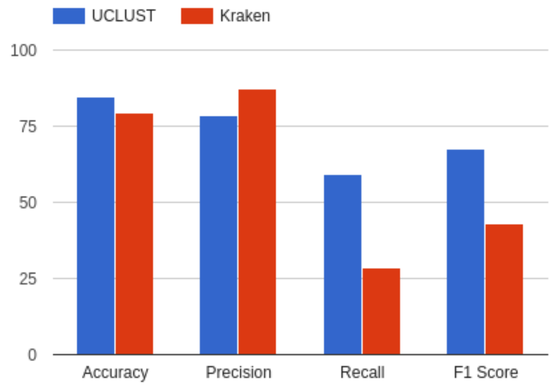


Fig. 2. This diagram shows the results for the comparison between UCLUST and Kraken when classifying ”Encephalopathy” versus all other classes. UCLUST had an accuracy and F1-Score of 84.53% and 67.44% respectively. Kraken didn’t perform as well, with an accuracy of 79.56% and an F1-Score of 43.08%.

Table I shows the run time of each representation in seconds. We had total of 1.3 million sequence reads. UCLUST unsupervised clustering took 180 seconds, while Kraken’s OTU database lookups took 120 seconds.

Figure 2 and Table II show the classification performance of UCLUST and Kraken in distinguishing patients in the ”Encephalopathy” class versus the other classes. UCLUST correctly predicted the clinical phenotype 84.53% of the time, and had a precision of 78.38% and recall of 59.18% on the held-out set. LOOCV results were slightly better for each metric than results on the test set by about 4-8%. Kraken had an accuracy of 79.56% with an even larger disparity between precision and recall (87.5% for the former and 28.57% for the latter), which led to a relatively poor F1-score of 43.08%. LOOCV results for Kraken were worse by 1.38% for accuracy, 17.19% for precision, 4.11% for recall, and 6.79% for F1-Score. As the results show, UCLUST generally performed better as a feature representation, with 4.97% higher accuracy and 24.36% higher F1-score.

Table III displays the performance of UCLUST-based classification with regards to training three one-versus-one clinical phenotypic classifiers. The first comparison distinguished between the ”Encephalopathy” class and the ”No Encephalopathy” class, effectively determining whether or not a patient who had liver cirrhosis also had encephalopathy. Results were much improved when compared with the findings in table II. The accuracy, precision, recall, and F1-score were all 86-90%. LOOCV results were worse by about 4-8% per metric. The second comparison distinguished between the ”Encephalopathy” class and the ”Control” class, effectively determining whether a patient had encephalopathy caused by liver cirrhosis or had neither disease. This comparison had the best results of the ones we examined, with greater than 95% performance in all metrics and 100% recall. The cross validation results were about 1-2% worse for each metric except recall, which was still 100%. The third comparison distinguished between the ”No Encephalopathy and liver cirrhosis” class and the ”Control” class, effectively determining which among the patients lacking encephalopathy had liver cirrhosis. This comparison had an accuracy of 83.78% and precision of 82.61%, but had a 100% recall which led to a high F1-score of 90.48%. Cross validation results were better by 3.72% for accuracy, 4.35% for precision, and 2.37% better for F1-Score, with essentially the same recall. Apart from the accuracy and precision of the third comparison, these three pairwise comparisons generally performed better across the board than the results in Table II, with the second pairwise comparison having the best performance.

Table IV displays the performance of Kraken with regards to the same three pairwise comparisons described above. For the Encephalopathy vs. No Encephalopathy run, the accuracy was 67.92%, with 82.14% precision, 44.23% recall, and 57.50% F1-Score. Cross validation produced a precision that was 10.07% lower but largely the same accuracy and recall. The Encephalopathy vs. Control run produced good results, with 78.69% accuracy, 76.92% precision, 97.56% recall, and 86.02% F1-Score. LOOCV results mostly confirmed those results, except the precision, which was 4.22% higher. The No Encephalopathy vs. Control run had quite similar results to the Encephalopathy vs. Control run. The main pattern that appears to emerge from these results is that the recall for Kraken was generally far superior in the pairwise runs than in the original run, which increased the F1-Scores but resulted in similar accuracy and precision. The Encephalopathy vs. No Encephalopathy run was significantly worse than the other two.

Table V shows the results for the same setup as in Table II after we performed feature engineering. For both representations in Table II, the recall was significantly lower than the precision, indicating that there were many more false negatives than false positives. We thus engineered the features in the training set. Certain features in the training set were most strongly present in positive patients, and as such were correlated with positive examples. Some examples in the test set had slightly lower values for those features, and were thus labelled by the classifier as negative examples, when in reality they were positive. We aimed to influence some of these borderline false negative predictions by lowering the threshold for those positively correlated features that would lead to an example being predicted as positive. In order to reduce the threshold for those positively correlated features, we applied a multiplier of 0.9 to the values of each feature of every positive patient in the training set. This had the effect of causing some previously borderline negative predictions to become positive

TABLE I. COMPARATIVE RUN TIME OF UCLUST AND KRAKEN REPRESENTATIONS.

Representation	Time
UCLUST	180s
Kraken	120s

TABLE II. COMPARATIVE PERFORMANCE OF UCLUST AND KRAKEN REPRESENTATIONS.

Representation	Accuracy	Precision	Recall	F1-Score	LOOCV Accuracy	LOOCV Precision	LOOCV Recall	LOOCV F1-Score
UCLUST	84.53	78.38	59.18	67.44	88.26	83.22	67.39	74.47
Kraken	79.56	87.5	28.57	43.08	78.18	70.31	24.46	36.29

TABLE III. THREE PAIRWISE COMPARISONS USING UCLUST.

Comparison	Accuracy	Precision	Recall	F1-Score	LOOCV Accuracy	LOOCV Precision	LOOCV Recall	LOOCV F1-Score
Encephalopathy vs. No Encephalopathy	88.68	90	86.54	88.23	84.47	81.77	81.77	81.77
Encephalopathy vs. Control	96.72	95.35	100	97.62	94.26	93.2	100	96.48
No Encephalopathy vs. Control	83.78	82.61	100	90.48	87.5	86.96	99.59	92.85

TABLE IV. THREE PAIRWISE COMPARISONS USING KRAKEN.

Comparison	Accuracy	Precision	Recall	F1-Score	LOOCV Accuracy	LOOCV Precision	LOOCV Recall	LOOCV F1-Score
Encephalopathy vs. No Encephalopathy	67.92	82.14	44.23	57.50	68.94	72.07	44.20	54.79
Encephalopathy vs. Control	78.69	76.92	97.56	86.02	79.51	81.14	96.35	88.09
No Encephalopathy vs. Control	77.03	77.03	100	87.02	81.42	81.42	100	89.76

predictions, primarily reducing the number of false negatives and improving the recall scores, but also positively impacting most other metrics as well. Attempts to use a multiplier of less than 0.9 did not appear to further improve the results.

With the multiplier of 0.9 applied to the feature values for positive training examples, the recall increased by 4.09% for UCLUST and by 12.25% for Kraken. This was confirmed in the cross validation results as well, in which UCLUST's recall increased by 2.18% and Kraken's recall increased by 11.41%. Because of the improvement in recall, both representations also had similarly improved F1-Scores after feature engineering. Accuracy and precision remained generally consistent. Kraken's LOOCV precision increased by 5.55%. This suggests that Kraken's very high precision was tied to its lack of many positive phenotype predictions in general, resulting in a very low number of false positives. While Kraken improved more than UCLUST did due to feature engineering, UCLUST still performed better in all metrics.

Tables VI and VII display the results after the same feature engineering techniques were applied to the pairwise results in III and IV, respectively. UCLUST's No Encephalopathy vs. Control run saw an increase of 4.06% in accuracy, 3.75% in precision, and similar gains in its cross validation results for accuracy and precision. UCLUST's other two runs were largely unaffected, albeit with slightly better cross validation results. Kraken's Encephalopathy vs. Control run was greatly improved after feature engineering, with an 8.20% increase in accuracy and a 9.75% increase in precision, with similar increases in cross validation results. This more than offset a slight decrease in recall. The Encephalopathy vs. No Encephalopathy run had mostly unchanged results, albeit with much improved cross validation scores, and the No Encephalopathy vs. Control run was not significantly affected.

VI. CONCLUSION

We have demonstrated a computational method to predict clinical phenotypes using metagenomic sequence data with intermediate OTU representation using state-of-the-art binning and taxonomic profiling approaches. Our results showed that the UCLUST representation achieved 85.64% accuracy and 70.46% F1-Score. Kraken performed less well, with 80.66% accuracy and 53.34% F1-Score. For the 1.3 million sequence reads UCLUST ran in 180 seconds and Kraken for 120 seconds, indicating that both performed efficiently enough to feasibly use them with larger data sets.

We then used UCLUST to test three one-versus-one clinical phenotype classifiers, which generally produced stronger results than our original comparison of the ”Encephalopathy” class versus all other classes. In particular, the classification of ”Encephalopathy” versus ”Control” performed the best. This intuitively seems reasonable. Whereas the original comparison requires making distinctions between those who have both encephalopathy and liver cirrhosis, those who have only the latter, and those who have neither, this pairwise comparison focuses on the simpler distinction between patients who have both diseases and those who have neither. The other two pairwise comparisons also similarly simplify the classification problem. When Kraken was used for the three pairwise comparisons, it had the peculiar effect of drastically increasing the recall while maintaining roughly the same accuracy and precision. This is most likely due to the fact that the pairwise comparisons involved data sets that were less negatively-skewed and thus didn't cause the classifier to predict as negatively based on Kraken's results. This is combined with the relatively strong accuracy and precision numbers in Kraken's original run, versus its poor recall numbers that left plenty of room for improvement.

Our technique of applying a multiplier to feature values for positive examples produced the effect of improving the recall in most cases. These results were generally corroborated by similar improvements in the cross validation results. Another effect was that precision and accuracy results were also often improved after applying this technique. This intuitively makes sense, as switching false negatives to true positives also positively affects the accuracy and precision ratios. Overall,

TABLE V. COMPARATIVE PERFORMANCE AFTER FEATURE ENGINEERING.

Representation	Accuracy	Precision	Recall	F1-Score	LOOCV Accuracy	LOOCV Precision	LOOCV Recall	LOOCV F1-Score
UCLUST	85.64	79.49	63.27	70.46	88.54	82.58	69.57	75.52
Kraken	80.66	76.92	40.82	53.34	80.8	75.86	35.87	48.93

TABLE VI. THREE PAIRWISE COMPARISONS USING UCLUST AFTER FEATURE ENGINEERING.

Comparison	Accuracy	Precision	Recall	F1-Score	LOOCV Accuracy	LOOCV Precision	LOOCV Recall	LOOCV F1-Score
Encephalopathy vs. No Encephalopathy	88.68	90	86.54	88.23	86.35	83.24	85.08	84.15
Encephalopathy vs. Control	96.72	95.35	100	97.62	95.08	94.12	100	96.97
No Encephalopathy vs. Control	87.84	86.36	100	92.68	91.22	90.57	99.59	95.05

TABLE VII. THREE PAIRWISE COMPARISONS USING KRAKEN AFTER FEATURE ENGINEERING.

Comparison	Accuracy	Precision	Recall	F1-Score	LOOCV Accuracy	LOOCV Precision	LOOCV Recall	LOOCV F1-Score
Encephalopathy vs. No Encephalopathy	69.81	83.33	48.08	60.98	77.18	80.88	60.77	69.40
Encephalopathy vs. Control	86.89	86.67	95.12	90.70	87.30	88.15	96.88	92.31
No Encephalopathy vs. Control	77.03	77.78	98.25	86.82	82.77	82.53	100	90.43

this feature engineering process appeared to have a generally positive effect on most of the results, suggesting that similar techniques for improving classification results merit further investigation.

Based on our results demonstrating over 85% accuracy for the original run and even better results for some of the pairwise comparisons, we believe the use of machine learning classification techniques to predict clinical phenotypes merits continued research. There are many potential practical uses of such technology for medical purposes, including diagnostics and research into diseases and their relationship with the human microbiome.

A. Future Work

Kraken is developed to be fast, accurate, and precise, all of which we found to be true. However, it was not as accurate as a representation in comparison to UCLUST. Kraken's poor performance was mostly due to poor matching of k-mers in the reads to taxonomic classes in its database. Often, a read was only a one or two percent match to the class that Kraken labeled it as. This may be due in part to database constraints. Kraken's full database is very computationally expensive to download and build. We used a 4 GB custom subset of the full 160 GB database. It is possible that the full database would have led to improved performance, and this is a future research question.

There is also an opportunity to use a much larger set of sequence reads. Having more training examples for the classifier would lead to a stronger and more generalizable model for predicting clinical phenotypes. The use of other annotation and classification methods could also be explored. Finally, a limited set of feature selection and scaling techniques were explored, and future research could investigate further how such techniques could be used to improve predictions.

ACKNOWLEDGMENT

The authors would like to extend our gratitude towards the Office of Research Computing at George Mason University.

REFERENCES

[1] P. J. Turnbaugh, R. E. Ley, M. Hamady, C. M. Fraser-Liggett, R. Knight, and J. I. Gordon, "The human microbiome project." *Nature*, vol. 449, no. 7164, pp. 804–810, Oct. 2007. [Online]. Available: http://dx.doi.org/10.1038/nature06244

[2] F. Backhed, R. E. Ley, J. L. Sonnenburg, D. A. Peterson, and J. I. Gordon, "Host-Bacterial mutualism in the human intestine," *Science*, vol. 307, no. 5717, pp. 1915–1920, 2005. [Online]. Available: http://www.sciencemag.org/cgi/content/abstract/307/5717/1915

[3] E. K. Costello, C. L. Lauber, M. Hamady, N. Fierer, J. I. Gordon, and R. Knight, "Bacterial Community Variation in Human Body Habitats Across Space and Time," *Science*, vol. 326, no. 5960, pp. 1694–1697, 2009. [Online]. Available: http://www.sciencemag.org/cgi/content/abstract/326/5960/1694

[4] J. Qin et al., "A human gut microbial gene catalogue established by metagenomic sequencing," *Nature*, vol. 464, no. 7285, pp. 59–65, Mar 2010.

[5] C. R. Woese and G. E. Fox, "Phylogenetic structure of the prokaryotic domain: the primary kingdoms." *Proc Natl Acad Sci U S A*, vol. 74, no. 11, pp. 5088–5090, 1977.

[6] R. Edgar, "Search and clustering orders of magnitude faster than blast," *Bioinformatics*, vol. 26, no. 19, pp. 2460–2461, 2010.

[7] Kraken: ultrafast metagenomic sequence classification using exact alignments. [Online]. Available: http://genomebiology.com/2014/15/3/R46

[8] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.

[9] P. Hugenholtz and G. Tyson, "Microbiology: metagenomics," *Nature*, vol. 455, no. 7212, pp. 481–483, 2008.

[10] A. Charuvaka and H. Rangwala, "Evaluation of short read metagenomic assembly," *BMC genomics*, vol. 12, no. Suppl 2, p. S8, 2011.

[11] J. Petrosino, S. Highlander, R. Luna, R. Gibbs, and J. Versalovic, "Metagenomic pyrosequencing and microbial identification," *Clinical chemistry*, vol. 55, no. 5, pp. 856–866, 2009.

[12] G. W. Tyson and P. Hugenholtz, "Microbiology: Metagenomics," *Nature Reviews*, vol. 455, pp. 481–483, Sep 2008.

[13] P. Schloss, S. Westcott, T. Ryabin, J. Hall, M. Hartmann, E. Hollister, R. Lesniewski, B. Oakley, D. Parks, C. Robinson *et al.*, "Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities," *Applied and environmental microbiology*, vol. 75, no. 23, pp. 7537–7541, 2009.

[14] P. Schloss and J. Handelsman, "Introducing dotur, a computer program for defining operational taxonomic units and estimating species richness," *Applied and environmental microbiology*, vol. 71, no. 3, pp. 1501–1506, 2005.

[15] Y. Sun, Y. Cai, L. Liu, F. Yu, M. Farrell, W. McKendree, and W. Farmerie, "Esprit: estimating species richness using large collections of 16s rrna pyrosequences," *Nucleic Acids Research*, vol. 37, no. 10, pp. e76–e76, 2009.

[16] Z. Rasheed, **Huzefa Rangwala**, and D. Barbara, "Lsh-div: Species diversity estimation using locality sensitive hashing," in *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. Philadelphia, USA: IEEE, October 2012, pp. 1–6, acceptance Rate: **59/299 = 19.93%**.

[17] W. Li and A. Godzik, "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, 2006. [Online]. Available: http://bioinformatics.oxfordjournals.org/content/22/13/1658.abstract

[18] X. Hao, R. Jiang, and T. Chen, "Clustering 16s rrna for otu prediction: a method of unsupervised bayesian clustering," *Bioinformatics*, vol. 27, no. 5, pp. 611–618, 2011. [Online]. Available: http://bioinformatics.oxfordjournals.org/content/27/5/611.abstract

[19] Z. Rasheed and H. Rangwala, "Mc-minh: Metagenome clustering using minwise based hashing," in *SIAM International Conference in Data Mining (SDM)*. Austin, TX: SIAM, May 2013.

[20] A. Brady and S. L. Salzberg, "Phymm and phymmbl: metagenomic phylogenetic classification with interpolated markov models," *Nat Meth*, vol. 6, no. 9, pp. 673–676, 2009. [Online]. Available: http://dx.doi.org/10.1038/nmeth.1358

[21] Z. Liu, T. Z. DeSantis, G. L. Andersen, and R. Knight, "Accurate taxonomy assignments from 16S rRNA sequences produced by highly parallel pyrosequencers," *Nucl. Acids Res.*, vol. 36, no. 18, pp. e120–, 2008. [Online]. Available: http://nar.oxfordjournals.org/cgi/content/abstract/36/18/e120

[22] J. R. Cole, B. Chai, R. J. Farris, Q. Wang, S. A. Kulam, D. M. McGarrell, G. M. Garrity, and J. M. Tiedje, "The ribosomal database project (rdp-ii): sequences and tools for high-throughput rrna analysis." *Nucleic Acids Res*, vol. 33, no. Database issue, pp. D294–D296, Jan 2005. [Online]. Available: http://dx.doi.org/10.1093/nar/gki038

[23] J. R. Cole, B. Chai, R. J. Farris, Q. Wang, A. S. Kulam-Syed-Mohideen, D. M. McGarrell, A. M. Bandela, E. Cardenas, G. M. Garrity, and J. M. Tiedje, "The ribosomal database project (rdp-ii): introducing myrdp space and quality controlled public data." *Nucleic Acids Res*, vol. 35, no. Database issue, pp. D169–D172, Jan 2007. [Online]. Available: http://dx.doi.org/10.1093/nar/gkl889

[24] J. R. Cole, Q. Wang, E. Cardenas, J. Fish, B. Chai, R. J. Farris, A. S. Kulam-Syed-Mohideen, D. M. McGarrell, T. Marsh, G. M. Garrity, and J. M. Tiedje, "The ribosomal database project: improved alignments and new tools for rrna analysis." *Nucleic Acids Res*, Nov 2008. [Online]. Available: http://dx.doi.org/10.1093/nar/gkn879

[25] T. Z. DeSantis, P. Hugenholtz, N. Larsen, M. Rojas, E. L. Brodie, K. Keller, T. Huber, D. Dalevi, P. Hu, and G. L. Andersen, "Greengenes, a Chimera-Checked 16S rRNA Gene Database and Workbench Compatible with ARB," *Appl. Environ. Microbiol.*, vol. 72, no. 7, pp. 5069–5072, 2006. [Online]. Available: http://aem.asm.org/cgi/content/abstract/72/7/5069

[26] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman, "Basic local alignment search tool," *Journal of Molecular Biology*, vol. 215, pp. 403–410, 1990.

[27] S. G. Tringe, C. von Mering, A. Kobayashi, A. A. Salamov, K. Chen, H. W. Chang, M. Podar, J. M. Short, E. J. Mathur, J. C. Detter, P. Bork, P. Hugenholtz, and E. M. Rubin, "Comparative Metagenomics of Microbial Communities," *Science*, vol. 308, no. 5721, pp. 554–557, 2005. [Online]. Available: http://www.sciencemag.org/cgi/content/abstract/308/5721/554

[28] D. Huson, A. Auch, J. Qi, and S. Schuster, "Megan analysis of metagenomic data," *Genome Res*, vol. 17, no. 3, pp. 377–386, 2007.

[29] M. Horton, N. Bodenhausen, and J. Bergelson, "MARTA: a suite of Java-based tools for assigning taxonomic status to DNA sequences." *Bioinformatics*, vol. 26, no. 4, pp. 568–569, 2010.

[30] F. Meyer, D. Paarmann, M. D'Souza, R. Olson, E. Glass, M. Kubal, T. Paczian, A. Rodriguez, R. Stevens, A. Wilke, J. Wilkening, and R. Edwards, "The metagenomics rast server - a public resource for the automatic phylogenetic and functional analysis of metagenomes," *BMC Bioinformatics*, vol. 9, p. 386, 2008.

[31] F. E. Angly, D. Willner, A. Prieto-Dav, R. A. Edwards, R. Schmieder, R. Vega-Thurber, D. A. Antonopoulos, K. Barott, M. T. Cottrell, C. Desnues, E. A. Dinsdale, M. Furlan, M. Haynes, M. R. Henn, Y. Hu, D. L. Kirchman, T. McDole, J. D. McPherson, F. Meyer, R. M. Miller, E. Mundt, R. K. Naviaux, B. Rodriguez-Mueller, R. Stevens, L. Wegley, L. Zhang, B. Zhu, and F. Rohwer, "The gaas metagenomic tool and its estimations of viral and microbial average genome size in four major biomes," *PLoS Comput Biol*, vol. 5, no. 12, p. e1000593, 12 2009.

[32] A. C. McHardy, H. G. Martin, A. Tsirigos, P. Hugenholtz, and I. Rigoutsos, "Accurate phylogenetic classification of variable-length dna fragments," *Nat Meth*, vol. 4, no. 1, pp. 63–72, 2007. [Online]. Available: http://dx.doi.org/10.1038/nmeth976

[33] H. Teeling, J. Waldmann, T. Lombardot, M. Bauer, and F. Glockner, "Tetra: a web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in dna sequences," *BMC Bioinformatics*, vol. 5, p. 163, 2004.

[34] A. L. Delcher, D. Harmon, S. Kasif, O. White, and S. L. Salzberg, "Improved microbial gene identification with glimmer," *Nucleic Acid Research*, vol. 27, no. 23, pp. 4436–4641, 1998.

[35] Svmlight support vector machine. [Online]. Available: http://svmlight.joachims.org/

[36] R. Edgar, "Search and clustering orders of magnitude faster than blast," *Bioinformatics*, vol. 26, no. 19, pp. 2460–2461, 2010.

[37] Kraken taxonomic sequence classification system operating manual. [Online]. Available: https://ccb.jhu.edu/software/kraken/MANUAL.html

[38] A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. [Online]. Available: http://www.xrce.xerox.com/content/download/16594/118473/file/xrce_eval.pdf

[39] On-campus research computing. [Online]. Available: http://orc.gmu.edu/research-computing/argo-cluster/argo-hardware-specs/