Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Date: June 5th, 2018
CS229 Report

CS229 Project Report

Phenotype Prediction Using Metagenomic Reads

Renju Liu [1], Sonali Garg [2] and Shweta Sood [3]

UID: 204760024[1], and UID: 104944076 [2], and UID: 905029230[3]

*To whom correspondence should be addressed.

Associate Editor: Wei Wang

Abstract

Motivation: Microbiome is usually used to understand and analyze the genomes of the environment. In this project, we are using the DNA genome sequences to analyze and to predict the possible phenotypes.
Results: In this project, we tried four different methods, on multiple datasets including cirrhosis, Type-2 Diabetes, obesity, etc. to identify and classify different kinds of diseases. After running our experiments using our methods, we obtained the best prediction results with accuracy up to 91% for IBD dataset using neural network classifier. We also compare our results with the the best results of other recent papers as baseline, and we showed that our methods have outperformed the methods in the original papers on the same datasets.
Availability: We have zipped all of our source code as well as a README file in the same submission of this report. We also indicate the datasets we use for this project, but because the data size is too large, we do not submit the datasets with this report. instead, we have included features and preprocessed data fed to classifiers.
Supplementary information: Supplementary data are available at *Bioinformatics* online.

	MetAML	Our Method
Features	1.Species abundances 2.Strain-specific marker presence	1.Species abundances 2.Strain-specific marker abundances 3.Combination of 1,2
Feature Extraction	MetaphlAn2	MetaphlAn2
Pre-processing	Normalization across entire data	1.Normalization across every sample 2.Binning on normalized values on a logarithmic scale based on the abundance distribution 3.Fill up images
Classifier	SVM, RDF, Elastic Net, Lasso	NN, CNN, RDF, XgBoost, SVM

Table 1. Comparison of our work with MetAML (6)

1 Introduction

Metagenomics have been widely studied in various ways. Generally speaking, metagenomics are the study of the analysis of microbial DNAs from the environment. By using tools to analyze these DNAs, it might reveal us unknown microbes or biomarkers, which could possibly cause an expression in individuals. For example, by analyzing the metagenomics reads from lung cancer patients, we can identify DNA pieces that cause such cancer.

Microbiome is useful to analyze and understand the cause of an individual phenomenon. Although we might not understand what has exactly happened for a specific piece of DNA sequences, we can guess the possible diseases of some DNA features expressed by the pieces of DNA sequences. More specifically, the human microbiome is crucial for human health and for the development and maintenance of the immune system and for several metabolic activities. The potential use of the microbiome as a diagnostic tool is a promising line of investigation. Hence, we want to assess the prediction power of metagenomic data in linking the gut microbiome with diseases such as obesity, type-2 diabetes (6).

In this project, we use metagenomic reads to make the predictions for phenotype. In order to explore the possible methodologies to make the phenotype prediction. We have investigated several related papers that

Dataset Name	Body Site	Disease	Positive Samples	Negative Samples
Cirrhosis	Gut	Liver Cirrhosis	118	114
Colorectal	Gut	Colorectal Cancer	48	73
IBD	Gut	Inflammatory Bowel Diseases	25	85
Obesity	Gut	Obesity	164	89
T2D	Gut	Type 2 Diabetes	170	174
WT2D	Gut	Type 2 Diabetes	53	43

Table 2. Datasets used for the experiment

we can potentially borrow the ideas from. MetAML (6) for metagenomic-based prediction tasks on the same six datasets as ours. This paper assessed viability of metagenomic data for various tasks like disease prediction, and non-disease classification like gender discrimination, body-site prediction etc. We focus on disease prediction. We will mostly use that paper as the baseline comparison for our results. Another paper (5) demonstrated how to use CNNs for metagenomic data and specifies a unique representation of the data as images. We show the comparison of our results with the results obtained in this paper too in Table 1. (7) exploits phylogenetic structure in microbial taxa. CAMIL (2) proposed a pipeline that requires no reference datasets for taxonomic assignments. One of the drawback of using features from MetaPhlAn2 (species-level relative abundances and abundances of strain-specific markers) is that having such features in this form is dependent on reference datasets of marker genes that need to be found from known genomes. This means that we could fail to retain information that could help in disease classification if a large number of sequences are new.

We use four different methods ranging from different feature extraction methods to different prediction pipelines such as clustering and assembly with multiple instance learning. We explore a method that has a better data preprocessing methods such as using normalization across each sample. We also explore neural network based approaches to further make an improvement.

We run and tested our methods with the comparison against the baseline methods. The datasets we use target diseases, namely, Cirrhosis, Colorectal, Inflammatory Bowel Diseases, Type-2 Diabetes, Obesity. The description of the datasets is detailed in 2. Our results have shown that we obtained as high as 91% accuracy. This report is organized as into the following: we discuss the methods and show the results from the datasets in Section 2. We present our final results in Section 3. We analyze and discuss the possible causes of the results in Section 4. Finally, we conclude our work in Section 5.

2 Approaches

In this Section, we discuss several different methods that we tried and optimized. We also show the corresponding results for each different method. The datasets we use to test our methods are shown in Table 2.

Dataset Name	Best Classifier (among RDF, XgBoost)	Accuracy (%)	MetAML (Best classifier %)
Cirrhosis	RDF	87.32	87.7
Colorectal	XgBoost	**83.78**	80.5
IBD	XgBoost	**91.2**	80.9
Obesity	RDF	64.94	64.4
T2D	RDF	**69.52**	66.4
WT2D	XgBoost	**79.31**	70.3

Table 3. Results of applying RDF and XgBoost after normalization

Dataset Name	Best Classifier (among RDF, XgBoost)	Accuracy (%)	MetAML (Best classifier %)
Cirrhosis	RDF	**88.73**	87.7
Colorectal	XgBoost	78.38	80.5
IBD	XgBoost	**88.24**	80.9
Obesity	RDF	64.94	64.4
T2D	RDF	**73.3**	66.4
WT2D	RDF	**82.76**	70.3

Table 4. Results of applying RDF and XgBoost after binning

2.1 Method I

The first intuition was to try a different preprocessing technique with the best performing classifiers - Random Decision Forests (RDF), and Extreme Gradient Boosting (XgBoost) reported on the same datasets as ours. The features used for this method were species abundances. Before performing classification, we tried two preprocessing techniques: 1) Normalization - For each sample, species abundance is represented as a real number and the total abundance of all species sums to 1. 2) Binning on normalized values on a logarithmic scale based on the abundance distribution. The intuition of using binning was that similar values of species abundances should be clubbed together.

2.1.1 Results

We try Random Forests and XgBoost as the classifiers for this method. We obtained the results as indicated in Table 3 and Table 4.

We can see from the tables that normalization and binning are effective preprocessing techniques. Just with the help of simple preprocessing and keeping the same classifiers, we observed appreciable jumps in accuracy for most of the datasets. IBD dataset saw close to 11% jump in accuracy after normalization and WT2D saw 9% jump. The jump for WT2D was more higher after binning - 12%. T2D dataset saw 7% jump after binning. This shows that simple techniques towards effective feature representation are important to explore before applying a classifier.

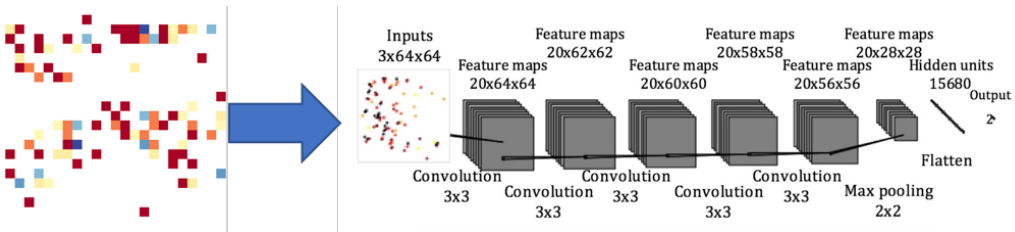


Fig. 1: The image on the left shows a fill-up image for Cirrhosis Dataset sample. The image on the right shows the architecture of CNN after the image on the left is passed to the network.

Our Method	Best Classifier of MetAML (RDF)	CNN
89.8%	87.7%	89.1%

Table 5. Results of Neural Network on Cirrhosis Dataset

	Healthy predicted	Disease predicted
Healthy actual	34	3
Disease actual	4	28

Table 6. Confusion Matrix for Cirrhosis Dataset

Our Method	Best Classifier of MetAML (RDF)	CNN
91.3%	80.9%	83.6%

Table 7. Results of Neural Network on IBD Dataset

	Healthy predicted	Disease predicted
Healthy actual	14	1
Disease actual	2	6

Table 8. Confusion Matrix for IBD Dataset

2.2 Method II

2.2.1 Convolutional Neural Network (CNN)

Literature Survey of the problem revealed Deep Learning based solutions have been proposed for this problem. We implemented the approach presented in (4) . We used species abundances as features and followed the following preprocessing pipeline:

- Normalization: For each sample, species abundance is represented as a real number and the total abundance of all species sums to 1.
- Binning on normalized values on a logarithmic scale based on the abundance distribution. A set of colors is chosen and applied to different bins such that bins with similar height (frequency) have the same color.
- Fill-up images are created by arranging abundance values into a matrix in a left-to-right order by row. The image is square and empty bottom-left of the image are set to zero (white). These fill-up images had features arranged in phylogenetic order. For phylogenetic-sorting, the features which are bacterial species are arranged based on their taxonomic annotation ordered increasingly by the concatenated strings of their taxonomy (i.e. phylum, class, order, family, genus and species).

Fig. 1 shows an example of the pipeline to feed the image to CNN network.

Results WT2D dataset gave 82.7% result with this method. However, the results of all the other datasets were close to 60%. Seemingly, a deep CNN architecture doesn't generalize well for datasets of small sizes and large variance The paper did report trying a lot of variations of CNN. While fine tuning our architecture, we realized that smaller architectures were performing better. This gave rise to our next set of experiments with neural networks.

2.2.2 Two Layer-Fully Connected Neural Network

Since shallower networks seemed to perform better, we used a small 2 layer fully connected neural network on Cirrhosis Dataset. The features were normalized species abundances. As opposed to the CNN, this network trains in a minute. The network was trained for 200 epochs at a learning rate of 0.1 with Adam optimizer.

Results As shown in Table 5, we can see that the fully connected neural network has a comparable performance with the CNN results of the paper (4). This appreciable performance comes at a much lower cost in terms of time required for training.

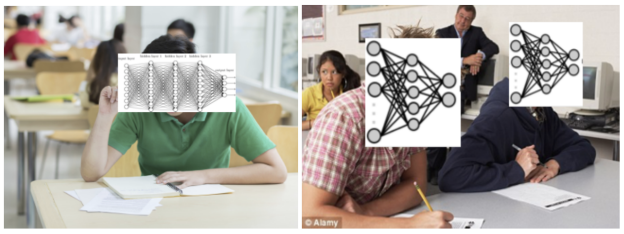
2.2.3 Preventing Overfitting on Unbalanced Datasets

Next, we tried the same approach for the IBD dataset. The same 2-layer fully connected neural network faired poorly. This was because the dataset was highly unbalanced (25 positive samples, 85 negative samples). As a result of this, there was overfitting on the training data. To overcome the same and to induce generalizability in the model, we fed a random batch size of only 7 samples of each class in every epoch and trained the model for just 100 epochs. This prevented overfitting as shown by the results.

Results Table 7 shows the results. Our method outperformed both the best classifier of MetAML and the CNN paper. This implies overfitting can be prevented by epoch thresholding and keeping randomly sampled small batch sizes.

2.2.4 Neural Network Ensembles

The same neural network architecture was now tried on the Obesity Dataset. The accuracy was very poor. No matter what we tried, the global minima of the problem was at 57%. This implied two things:



(a) (b)

(a) Does not realize that it can communicate with other Neural Networks to solve the exam, so has to work harder to score !
(b) Leverages on the fact that cheating and communication is allowed during the exam, so don't have to work that hard !

Fig. 2: Intuition of the approach.

Our Method	Best Classifier of MetAML (RDF)	CNN
70.6%	64.4%	66%

Table 9. Results of Neural Network on Obesity Dataset

	Healthy predicted	Disease predicted
Healthy actual	36	14
Disease actual	8	19

Table 10. Confusion Matrix for Obesity Dataset

Method	**Accuracy**
MetAML paper (6)	66.4%
Case 1 before feature selection	66.95%
Case 2 before feature selection	69.37%
Case 1 after feature selection	69.60%
Case 2 after feature selection	71.23%
Combining both cases after feature selection	73%

Table 11. Accuracy of all Cases

1. Trying Non-Gradient Based Approaches - Which yielded 65% accuracy as shown in Method I.
2. Inducing Generalizability in Gradient Based Approaches so that we can reduce errors by keeping many weak learners.

The line of thought expressed in point 2, is inspired from the intuition expressed in Fig. 2.

We have a breadth learner and a depth learner. The breadth learner learns about all the types of test cases it can see, though on a very broad level. We also have a depth learner that goes into the depth of topics. If these two learners collaborate with each other, it can improve the overall accuracy of the system. This idea is inspired from collaboration scenarios seen in examination. If we have a student doing the exam all by himself, he will have to work harder to score during the exam. Instead, if we have two students that can communicate during the exam, they won't have to work that hard. One of them would just briefly skim all topics to be tested and the other would go into depth of specific topics. Now, all the questions that the breadth learner ain't able to solve will be given to the depth learner. This would ensure a better overall accuracy of the system.

Results As shown in Table 9, using the ensemble of two shallow networks (10 nodes on a layer), we obtained an accuracy of 70.6%. Moreover, the training time of the system was less than 15 seconds which is a great speedup over a traditional CNN architecture. This is 6% jump over the MetAML accuracy and 4% accuracy over CNN (4) paper. This based on the idea, "We want our ensemble to consist of highly correct networks that disagree as much as possible". Theoretically, if the average error rate for a pattern is less than 50% and the networks in the ensemble are independent in the production of their errors, the expected error for that pattern can be reduced to zero as the number of networks combined goes to infinity.

Another theoretical approach - Unlearning: Instead of creating a generalized and a specialized network, we could create two specialized networks. We first train a network on the entire training data and take account of correct and incorrect classifications. Next, we delete this original network and use two specialized networks, one for the correct and the other for the incorrect classifications. Thus, both the networks specialize on different types of data. Whichever network gives better accuracy during it's training is used first. If it's confidence on a certain test example is less, it is passed to the second network.

2.3 Method III

Having tried different preprocessing techniques and classification methods using species abundances as our features, in this part we looked at using strain-specific marker abundances as or features. Our hypothesis was that complex diseases are associated with the presence of specific strains or subspecies rather than only species-level abundances (6). The Type-2 Diabetes dataset was used in the experiments, with Random forests as the classifier. Firstly, we simply run the MetaML tool (6) in 2 cases:

- Case 1: Using species abundances as features
- Case 2: Using strain-specific marker abundances as features

The tool (by default) reports results using 20 runs of 10 fold cross validation, so we kept that the same across the 2 cases for a direct comparison to test the veracity of our hypothesis. We have used this throughout Method III.

Results Comparing our results(all results for method III have been presented in Table 11), it was clear that our hypothesis was correct. However, we notice that the number of features for Case 1 were 1444 and for Case 2 were 127,247. This meant that the time taken for running Case 2 was extremely high as compared to Case 1. Also, since number of

	Healthy Predicted	Disease Predicted
Healthy Actual	2391	1089
Disease Actual	1184	2216

Table 12. Confusion matrix of Case 1 before feature selection.

	Healthy Predicted	Disease Predicted
Healthy Actual	2427	1053
Disease Actual	1056	2344

Table 13. Confusion matrix of Case 2 before feature selection

	Healthy predicted	Disease predicted
Healthy actual	1246	494
Disease actual	552	1148

Table 14. Confusion matrix for Case 1 after feature selction

	Healthy predicted	Disease predicted
Healthy actual	1215	525
Disease actual	464	1236

Table 15. Confusion matrix for Case 2 after feature selection

	Healthy Predicted	Disease Predicted
Healthy Actual	1296	444
Disease Actual	483	1217

Table 16. Confusion matrix on combining Case 1 and 2 after feature selction

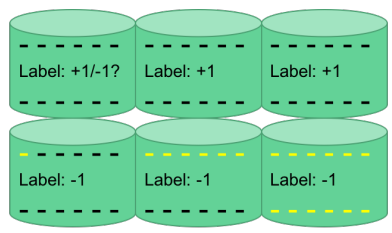


Fig. 3: Example of multiple instance learning labeling (2)

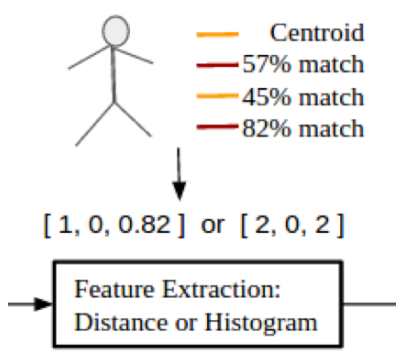


Fig. 4: Example of CAMIL feature generation methods (2)

features in Case 2 is pretty high, it meant that there was potentially a lot of noise in the features. Taking such a large number of features, specially combined with relatively very less samples, does not make much sense. We decided to resolve this through Feature selection.

We used the Feature importances found by RDF for this purpose. It is also known as "gini importance" or "mean decrease impurity". It represents the total decrease in node impurity averaged over all trees of the ensemble. The node impurity is weighted by the probability of reaching that node.

At this point, we did a sanity check. According to our results, s_Roseburia_intestinalis was the most important feature. We found other studies online stating a link between type 2 diabetes and this feature. Thus, verifying that were on the right track. Also, the complete list of important features can be found in the files submitted.

Now, on taking top 50 features for Case 1, we saw that our accuracy improved as shown in Table 11.

On taking top 100 features for Case 2, again our accuracy improved.

Thus, our hypothesis was correct, taking less features actually improved accuracy and time taken now is substantially less (60 times less). All this proves that feature selection is very important in this problem. Not only does it help us reduce time and memory used, we also improve our accuracy. Too many features just lead to more noise and overfitting.

Our next hypothesis was that, now that we had a reasonable number of features for both Case 1 and Case 2, the two cases could be combined efficiently with potential for even better results. So, we combined the important features from Case 1 and Case 2 with even better results as shown in Table 11.

Hence, our hypothesis was correct.

2.4 Method IV: Clustering and Assembly with Multiple Instance Learning (2)

Multiple Instance Learning is useful when in addition to classifying patient phenotype, we also want to identify individual parts of the microbiome (indicative of that genotype) for a better understanding of the disease. In other words, it trains the dataset with unlabeled reads but with labeled instances. As we can infer from the name, multiple instance learning contains several clusters, which are also known as labeled bags. Each labeled bag contains unlabeled instances. We might not be able to understand what the unlabeled instances mean, but they are the features of the data. This is also similar to deep neural network. In the output layer, we know that high probability is better than low probability, but we do not understand what the probability stands for. Fig. 3 shows a demonstration of multiple instance learning. In this example, what we've known is the label of each container, but we do not have an understanding of the genome sequences (dotted lines) of each label.

Based on multiple instance learning mechanism, clustering and assembly with multiple instance learning uses the group features of the metagenomic data. It can predict the disease state, also known as phenotype, of patients based on the clustered and assembly metagenomic data. Using this method could help us make a prediction on whether a given person's DNA has a specific type of disease without completely understanding his or her DNA sequences.

The pipeline of CAMIL (2) is shown in Fig. 5. Starting from FASTQ data, it first assembles the patients' DNA sequence information using SOAPDenov2 (3) algorithm. Then it combines the assembled FASTQ

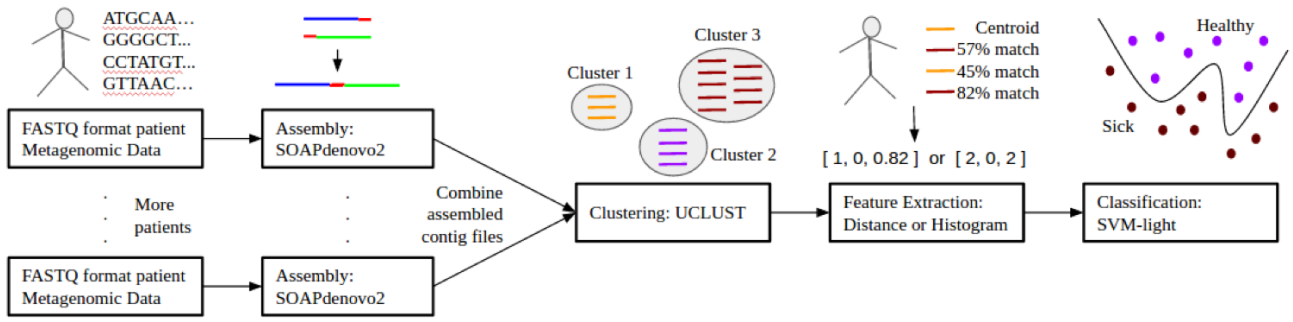


Fig. 5: Pipeline of Clustering and Assembly with Multiple Instance Learning (CAMIL) (2)

	SVM	Random Forest	Neural Network
Accuracy	76.11%	77.43%	71.90%
Precision	76.06%	78.25%	78.46%
Recall	99.71%	97.07%	86.51%
F1	86.29%	86.65%	82.29%

Table 17. Results of CAMIL

files into different clusters through UCLUST. The UCLUST algorithm clusters different sets of sequences. The cluster_fast and cluster_smallmem commands are based on UCLUST. A cluster is defined as the centroid or representative sequence. Every sequence in the cluster must have similarity about the centroid representative sequence. After using UCLUST, the same representative sequence of clusters will be determined, making it easier to extract features from them.

2.4.1 Feature extraction

One of the most important steps in this pipeline is feature extraction. After the original data gets clustered, the feature extraction mechanism needs to properly extract the features from each cluster, so that the classifiers can be applied to learn and make predictions of the disease. In this case, we used a "vocabulary-based" feature extraction method. More specifically, we use bag of words method. It first generates instance clustering, and then a feature vector is created to map the instance to the bags. Finally, the different classifiers will be used and applied to learn those features and make a prediction based on them. In our experiment, we used a simple SVM classifier as our baseline as a comparison with the implementation of the original paper. Moreover, we use random forest and neural networks as different classifiers to compare with our baseline case. We use both D-BoW and H-BoW as the feature extraction for our methods. D-BoW is the distance-based bag of words and H-Bow is the histogram of bag of words. The trade-offs between them are that while D-BoW is more time-efficient, H-BoW is more accurate. Fig. 4 shows an example of bag of words.

Results We run our method for T2D dataset with different classifiers of CAMIL. The T2D dataset for this experiment is not exactly the same as the datasets for the previous 3 methods while some of them are overlapped. Table 17 shows the results of CAMIL. It shows us that random forest outperforms other methods with the highest accuracy of 77%.

3 Final Results

Table 18 summarizes the best results of all methods on the six datasets. We can see that the Cirrhosis dataset sees a 2% jump with respect to the MetAML classifier using Neural Network. The IBD dataset sees a 11% jump using Neural Networks. Ensemble based Neural Networks see a 6% jump in obesity dataset. Method 4 gives a 12% jump on T2D dataset. RDF

Dataset	Our Method (%)	CNN(%)	MetAML (Best classifier %)
Cirrhosis	89.8 (NN after normalization)	89.1	87.7
Colorectal	83.78 (XgBoost after normalization)	74.2	80.5
IBD	91.3 (NN after normalization)	83.6	80.9
Obesity	70.6 (Ensemble based NN)	66	64.4
T2D	77.4 (Method IV)	62.6	66.4
WT2D	82.76 (RDF after binning)	58.9	70.3

Table 18. Summarizing best results of all methods on the six datasets

after binning yields a 12% higher accuracy on WT2D dataset. Colorectal dataset observes 4% jump using XgBoost after normalization. Overall, all our methods are giving higher accuracy compared to the CNN (4) paper and MetAML paper (6).

4 Discussion

We will discuss about our results in this section.

Looking at the results over the datasets, we can see that ensembling based architectures (RDF, Xgboost, Neural Networks) are helpful. This is because ensembles help in averaging out biases, reducing the variance. Hence, they are unlikely to overfit. This can also be seen as collaborative learning where, different networks in an ensemble become good at specific type of data. This improves the generalizability of the model. This was further seen in Method II, where we expounded on collaborative learning between a generalized model (breadth learner) and specialized model (depth learner). The idea is to increase the amount of mutual exclusion with respect to the examples these classifiers are trained upon. RDF and small sized neural networks perform much better for this data, both in terms of performance and time complexity.

Domain knowledge is important for any Machine Learning problem. This is true for this problem statement as well. Since species abundances imply how prominent that species is for representation of that sample, it made sense to use relative abundances such that the relative importances are captured. Our intuition was that for a disease, certain species would be more important and that should be captured before feeding to a classifier.

As is often the case in machine learning, we also notice that small (but sensible) preprocessing steps can make a huge impact. In Method I, we showed that normalization of dataset across samples helps more than normalization across entire dataset as it will emphasize relative abundances of the strains in a sample. Also, binning the species abundances on a logarithmic scale, so that their ranges matter more than specific values

was also shown to considerably improve accuracy. So, these theoretically small-seeming steps can be very important in practice.

The results reported in this paper are not just overall accuracies but confusion matrices also. The confusion matrices reveal a high sensitivity and high specificity. High sensitivity means the percentage of sick people who are correctly identified as having the condition. High specificity means the percentage of healthy people who are correctly identified as not having the condition. This becomes important in medical scenarios to provide immediate care to sick people and not providing unnecessary treatment / mistreatment to healthy people. So, we ensured that we were not only getting high accuracies (which can be easy to do by overfitting, especially in cases of imbalanced data) but good sensitivity and sensitivity too.

We have avoided using algorithms that rely on distance based metrics (like K Nearest Neighbors) in this project. Given the large feature space, the curse of dimensionality would surely come into play. As pointed out by Aggarwal (1), in high dimensions, a curious phenomenon arises: the ratio between the nearest and farthest points approaches 1, i.e. the points essentially become uniformly distant from each other. This phenomenon can be observed for wide variety of distance metrics, but it is more pronounced for the Euclidean metric. The premise of nearest neighbor search is that "closer" points are more relevant than "farther" points, but if all points are essentially uniformly distant from each other, the distinction is meaningless.

Another thing to note is that we also tried a method (Method IV) that requires no reference datasets for taxonomic assignments. As mentioned before, it removes the drawback of using MetaPhlAn2 (dependency on reference datasets of marker genes that need to be found from known genomes). Given that most microbes have not been laboratory-cultured and thus may not be known, it is a possibility that we failed to retain information that could help in disease classification due large number of new sequences. If trying different methods and preprocessing techniques does not yield good results on some dataset, then the technique suggested in Method IV could be highly useful.

Throughout this project, we have tried to consider not only accuracies but the time and memory required to achieve those accuracies. Using very shallow (2 layer) and ensembles of neural networks and extreme reduction in features in method III, We reduced the time taken in classifictaion step from hours to minutes.

The dataset size was small and had high variance. Thus, Deep Learning based approaches won't perform too well. They tend to overfit.

We saw in Method III that feature selection is highly useful in this problem. Firstly, it helps in reducing time and memory used. We were able to from around 1500 features to 50 features in Case 1 and more importantly, from around 127,000 features to 100 features in Case 2. Hence, running time was reduced more than 60 times.(Running time reduced from around 12 hours to 12 minutes). Secondly, this reduction in time was accompanied by an increase in accuracy also, making it even more impressive. Hence, we can say that high dimensional features like marker's abundances of the order of around 127K for a small dataset would only add more noise. Thirdly, it made combining species abundance and strain-specific marker abundance features viable, which led to even better accuracy. Epoch thresholding, random subsampling of small batch sizes helps in controlling overfitting for small datasets.

5 Conclusion

Through a combination of preprocessing, feature engineering and selection, and wide variety of classifiers we have been able to outperform recent papers like (6) and (4), our two baselines. We demonstrated the effectiveness of our methods on 5 different diseases (across 6 different datasets), which shows that they are generalizable and can possibly be applied to metagenomic data for other diseases. This shows that metagenomic data is indeed very useful in phenotype prediction. As part of the future work, following things need to be explored and improvised upon:

- Trying larger datasets - Currently the datasets available to us are of smaller sizes. Larger datasets may be useful in a more refined feature engineering and building robust classifiers.
- Alternatives to reduce feature extraction time - The amount of time required to extract features from MetaPhlAn2 is large. There is a need to explore other feature extraction tools/ methods that can optimize over this bottleneck.
- Perform non-disease classification: gender discrimination, body-site prediction.
- Subtractive assembly based approaches can be tried to find microbiomes that exist in healthy but not in diseased people as shown in (8).

Acknowledgements

We would like to thank Prof. Wei Wang's support for this project.

References

[1]Aggarwal, C. C., Hinneburg, A., and Keim, D. A. On the surprising behavior of distance metrics in high dimensional spaces. In *Proceedings of the 8th International Conference on Database Theory* (Berlin, Heidelberg, 2001), ICDT '01, Springer-Verlag, pp. 420–434.

[2]LaPierre, N., Rahman, M. A., and Rangwala, H. Camil: Clustering and assembly with multiple instance learning for phenotype prediction. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (Dec 2016), pp. 33–40.

[3]Luo, R., Liu, B., Xie, Y., Li, Z., Huang, W., Yuan, J., He, G., Chen, Y., Pan, Q., Liu, Y., Tang, J., Wu, G., Zhang, H., Shi, Y., Liu, Y., Yu, C., Wang, B., Lu, Y., Han, C., Cheung, D. W., Yiu, S.-M., Peng, S., Xiaoqian, Z., Liu, G., Liao, X., Li, Y., Yang, H., Wang, J., Lam, T.-W., and Wang, J. Soapdenovo2: an empirically improved memory-efficient short-read de novo assembler. *GigaScience 1*, 1 (Dec 2012), 18.

[4]Nguyen, T. H., Chevaleyre, Y., Prifti, E., Sokolovska, N., and Zucker, J. Deep learning for metagenomic data: using 2d embeddings and convolutional neural networks. *CoRR abs/1712.00244* (2017).

[5]Opitz, D. W., and Shavlik, J. W. Generating accurate and diverse members of a neural-network ensemble. In *Advances in Neural Information Processing Systems* (1996), MIT Press, pp. 535–541.

[6]Pasolli, E., Truong, D. T., Malik, F., Waldron, L., and Segata, N. Machine learning meta-analysis of large metagenomic datasets: Tools and biological insights. *PLOS Computational Biology 12*, 7 (07 2016), 1–26.

[7]Reiman, D., Metwally, A. A., and Dai, Y. Popphy-cnn: A phylogenetic tree embedded architecture for convolution neural networks for metagenomic data. *bioRxiv* (2018).

[8]Wang, M., Doak, T. G., and Ye, Y. Subtractive assembly for comparative metagenomics, and its application to type 2 diabetes metagenomes. *Genome Biology 16*, 1 (Nov 2015), 243.