Predictive Classifiability of Meta'omic Data Samples

Anthony Gray

Metagenomics AS.410.734.81 Final Project

Received on 12/17/19; revised on 01/15/20

ABSTRACT

This project examines and compares the performance of four machine learning classifiers on metagenomic and metatranscriptomic samples. Naïve Bayes, Support Vector Machines, Random Forests, and K-Nearest Neighbor Classifiers are applied using the protein family composition of an 'omic sample to determine if that sample came from a donor with Crohn's Disease or from a donor without Inflammatory Bowel Disease. All the algorithms were at least somewhat successful depending on the data preparation, with several combinations resulting classification accuracies higher than 90%. Overall, these algorithms had higher accuracies on the metagenomic data than on the metatranscriptomic data.

1 INTRODUCTION

The increasing availability of biological data has allowed researchers to discover relationships that would have been impossible to examine in the past. Genetic and genomic sciences have led to the identification of the causes and risk factors for many diseases and conditions. By sequencing an individual's DNA, it can be possible to better assess that person's risk for certain diseases, make lifestyle recommendations, and even safely tune the dosage of medication to their specific needs. However, not every condition is responsive to this type of investigation. Many conditions are complex, arising from the interaction of the individual's genes and their environment over time. While there may be risk factors for such a condition, each factor's contribution to the overall risk is small, and determining appropriate treatments or interventions for such conditions is difficult or impossible in many cases.

Crohn's Disease is one such condition, a form of inflammatory bowel disease that has no known single cause (Zuo and Ng, 2018). Many factors have been linked to its etiology, including diet and the composition of the patient's gut microbiota. The gut microbiota, or microbial population of the gut, has also been implicated as a factor in many conditions, from cancer to obesity (Vivarelli et al., 2019), and a lot of work has been done attempting to describe it's behavior and to link that behavior to disease development.

Broadly speaking, there are two ways of characterizing a population of organisms. One option is to look at the total genetic content of the population, or it's metagenome. This information begins as sequences of DNA sampled from an environment, but these sequences can be assembled into genomes or genome fragments and used to predict the presence and function of the proteins produced in that environment. The metagenome encodes all the genes of every organism in the population and can be used to identify the presence of species that cannot be cultured under lab conditions. Another option is to investigate the population's metatranscriptome. While the metagenome is the total DNA sequence of a population, the metatranscriptome is the total RNA sequence. In other words, the metagenome contains everything that a population can do, and the metatranscriptome contains information on what the population is doing right now, or what was happening in the very recent past. Not all of an organism's genes are always translated and expressed, and the same is true of a population of organisms. It is possible that a disease state only arises from the expression of certain genes, and while those genes might be present in the metagenome at all times, their presence or absence would not be useful for identifying them as risk factors because they would appear in the metagenomes of healthy individuals as well. On the other hand, if the genes only appeared in the metatranscriptome of individuals with the disease, they might be easier to link to the disease.

Another feature of meta'omic data is that there is a lot of it. The total genetic content of a pond, or of an individual's gut contains a tremendous amount of information in each sample, and multiple samples are necessary for controlled studies or to investigate effects over time. Because of this, machine learning techniques lend themselves to meta'omic analysis. In broad terms, machine learning is a programming paradigm where, rather than explicitly providing the information necessary for a task, a programmer writes the instructions for finding that information. With a flexible algorithm and enough training data, it is possible for machine learning algorithms to find patterns in huge datasets that would be impractical, if not impossible for a human to do by hand.

There have been many studies using machine learning to examine meta'omic data. Many such studies have attempted to predict whether a sample came from a healthy donor, or a donor with a disease. For example, Asif, Martiniano, Vicente, & Couto (2018) used gene ontology terms to identify genes associated with autism, and other studies have attempted to link environment and host phenotype (Qu, Guo, Liu, Lin, & Zou, 2019). However, many of these studies use operational taxonomic units (OTUs), or the species composition for their predictions (Asgari, Garakani, McHardy, & Mofrad, 2018). OTU analysis can be computationally expensive, as well as unreliable for two reasons. The first is that OTUs do not necessarily correlate with species, and the second is that the species present do not necessarily determine how the microbiota behaves. Different species can perform different roles within the community of the microbiota, so it is often more informative to look at the functional, rather than the taxonomic composition of a population. Biomarkers can often lead to better predictions (Pasolli, Truong, Malik, Waldron, & Segata, 2016), and in some cases the functional profile of the metatranscriptome can account for

To whom correspondence should be addressed. agray31@jhu.edu

greater differences between individuals than that of the metagenome (Li, Hitch, Chen, Creevey, & Guan, 2019).

However, reliable associations of this type can be hard to make. Meta'omic data is expensive to produce, in terms of both time and money, so sample sizes in these studies are often small (Zhou & Gallins, 2019). Studies can also be difficult to control. Because microbial populations are highly sensitive to changes in their environment, the gut microbiota is affected by things like diet and medication, so these factors must be accounted for in order to draw solid conclusions. Finally, high-dimensional data can reduce the effectiveness of machine learning techniques, so simply adding hundreds of thousands of genes and other variables to a dataset can confound the analysis. The features under investigation for a study, even one aided by powerful algorithms, must be carefully selected (Bang et al., 2019).

To that end, this project will examine how the choice of metatranscriptomic data or metagenomic data affects the accuracy of machine learning classifiers. Metagenomic data contains more features, but the metatranscriptome data is often smaller and could hold different clues as to the etiology of a disease if the confounding variables can be controlled well enough. The actual performance of the algorithms is less important in this case than how they operate of different types of data drawn from the same individuals and processed in the same way. Additionally, the features under investigation will be the protein family composition of each meta'ome, as the protein families serve as a reasonable proxy for the functional profile of a population. The disease in question is Crohn's Disease, as it is known to be highly complex and related to the behavior of the gut microbiota. It is also relatively easy to investigate via stool sample, so the quantities of data necessary for successful machine learning classification are publicly available online.

2 DATA PREPARATION

A large dataset was required in order to investigate the potential differences in classifiability between metagenomic and metatranscriptomes samples. The dataset needed to have large amounts of both types of sample data as well as information regarding the disease status of the donor for each sample. Unfortunately, while there is a great deal of meta'omic data publicly available across many databases, much of it is only raw sequence data. Most of the data has not been processed, and most of what has been annotated does not have information regarding the donor's disease status.

An exception is the Inflammatory Bowel Disease Multi'omics Database (ibdmdb.org). The IBDMDB contains 'omic data from several large clinical studies, and provides the raw sequences, as well as annotated protein family and taxonomic composition files for each sample. Most crucially, the sample metadata is also available, and includes information regarding the patient's demographic details, disease status, and answers to other questions of clinical relevance. Most of this metadata was discarded, as only the patient's unique identifier, their disease status, and the type of 'omic data were relevant to this project.

One disadvantage to the IBDMDB dataset was that all of its sample files were stored in separate databases corresponding to their sub-project of origin. The first step was therefore to identify the download URL for each sample and add those URLs to the metadata file. The sample data were then downloaded, and the gene family abundance files extracted. These abundance files contained both the percentage abundance of each gene family in the sample, as well as the percentage abundance of each gene family across the species present in the sample. The species abundance data was discarded while the total abundance data was retained, as doing so reduced the size of each dataset by around 30%.

With the useful sample data extracted, the represented gene families were collected and counted across all samples in the dataset. Each dataset was then filtered in two ways. First, gene families that were not present in at least 99% of the samples were removed. Biological data is extremely noisy, and the meta'omic composition between individuals varies widely. For example, is unlikely that a gene family present on only one or two of several hundred Crohn's Disease patients is associated with the disease.

Second, the difference in abundance between control and disease samples was used to filter out gene families. The gene family counts across disease samples were subtracted from those of control samples, and all samples with Δ values greater than 1.5 standard deviations or less than 2.5 standard deviations from the mean Δ were discarded. These methods both reduced the number of gene families in the dataset by several orders of magnitude, though the reduction by percentile count was greater than that by variance in control/disease abundance. The resulting size of each dataset is shown in Table 1.

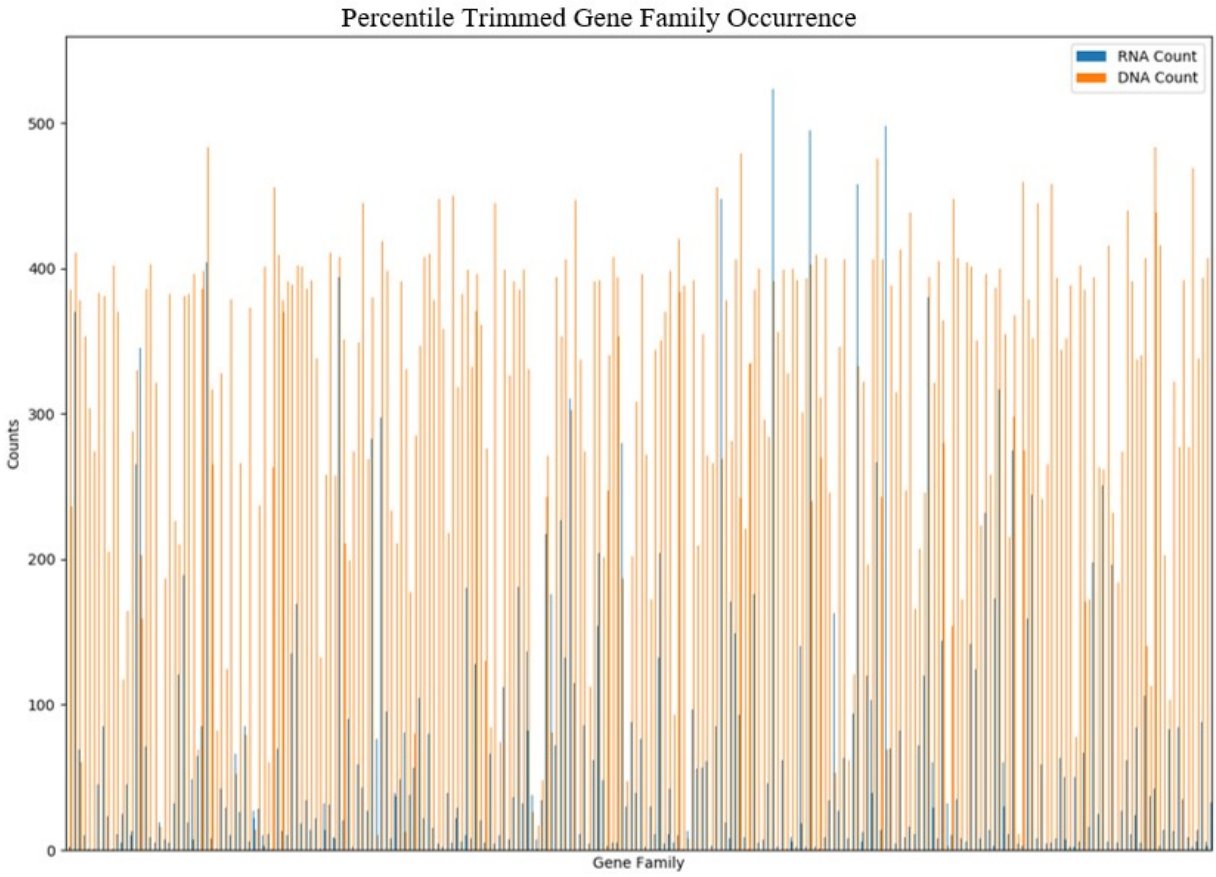
The counts of each gene family in each dataset were also plotted against each other, and while these plots do not show information regarding disease status, they do show that the metatranscriptomes and metagenomes have very different gene family compositions.

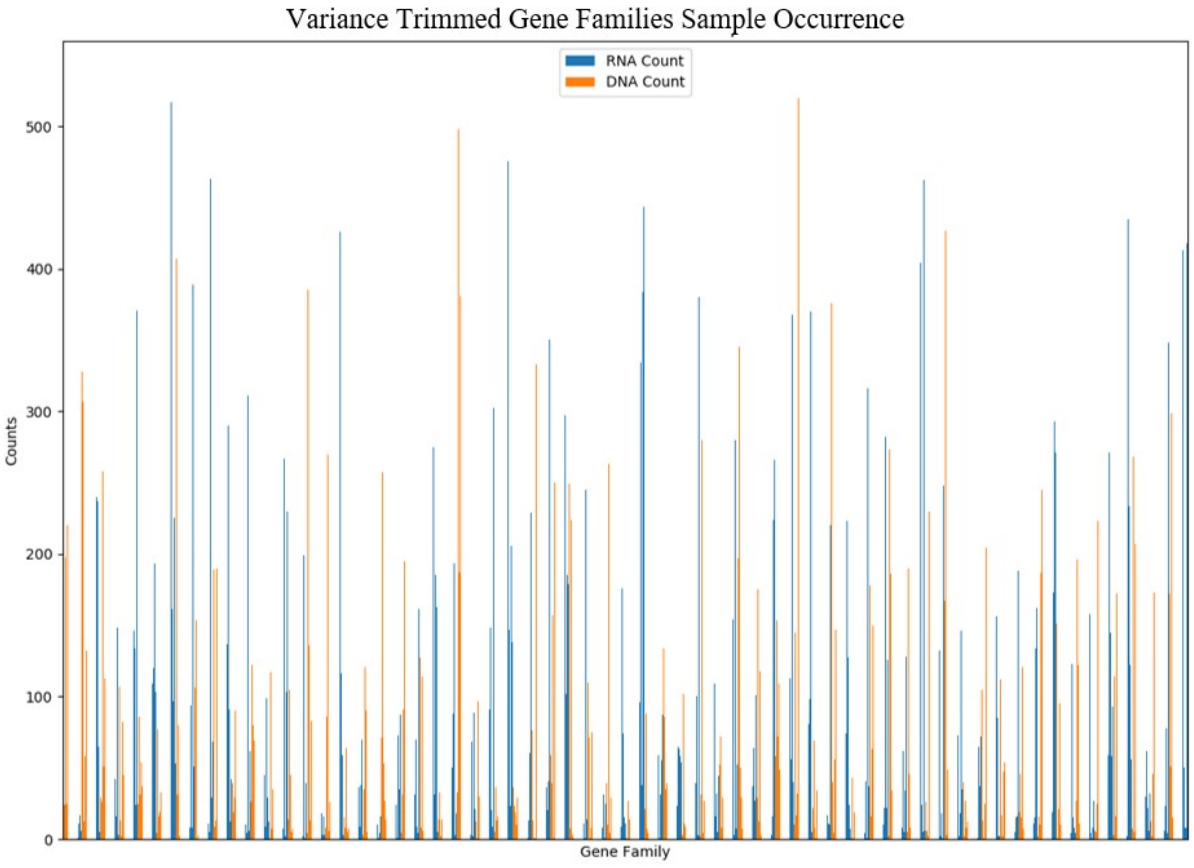
Table 1: Dataset Statistics	Sample Counts		Gene Family Counts		
	Control	**Crohn's Disease**	**Total**	**99th Percentile**	**Greatest Δ**
Metatranscriptome Data	187	337	823,297	7,657	18,695
Metagenome Data	364	599	1,636,945	15,132	56,214

Additionally, Hierarchical Clustering was performed on each dataset, though the dendrograms did not show any meaningful patterns regarding the disease state. The variance-filtered metagenome dendrogram was too large to be rendered at all. The dendrograms are included in the appendix.

With the samples as instances and the selected gene families as features, each dataset was assembled. Missing values were set to 0, and any abundance percentage less than 0.00001 was also set to 0. Columns containing the same value in each row were also dropped to further reduce the size of the dataset and remove potential noise.

Additionally, there were many more Metagenomic samples than Metatranscriptomic samples available, so the Metagenomic dataset was randomly sampled to match the size of the non-IBD and Crohn's Disease groups of the smaller dataset in order to control for the number and distribution of instances in each. This sampling was based on donor ID, and there were a few donors that occurred multiple times in the dataset due to providing samples at different times. This caused the size of the datasets to


Percentile Trimmed Gene Family Occurrence
RNA Count
DNA Count
Counts
Gene Family
0
100
200
300
400
500


Variance Trimmed Gene Families Sample Occurrence
RNA Count
DNA Count
Counts
Gene Family
0
100
200
300
400
500

vary by less than ten instances, but in the context of classification this was not significant.

Finally, each of these four datasets was processed with Principal Component Analysis or PCA (Pearson, 1901). PCA is a method of dimension reduction that seeks to re-orient the data onto fewer dimensions while maintaining as much of the variability of the data as possible. These new axes are known as principle components, and while they do not correspond to the features of the old dataset, each represents a merging of some of those features and can be used to perform classifications in the same way as the original dataset. The principal components themselves can also be ranked in order of importance, or how much of the datasets variation they contain. The PCs can then be visualized on a scree plot, allowing one to see which PCs are the most valuable and which can be safely discarded.

To proceed with PCA, each dataset was standardized, or oriented about the origin, the analysis was performed, and the scree plots generated. The standardized dataset was set aside for later classification, and a PCA dataset was generated from the PCs providing 95% of the original variance. This resulted in 12 datasets for classification, with their dimensions shown in Table 2. The scree plots were not particularly useful as they indicated a high number of low-significance PCs, but they are included in the appendix as well.

Table 2: Final Datasets	% Filtered Features	Δ Filtered Features	% PCA	Δ PCA
Metatranscriptome	535 x 7,657	535 x 18,695	535 x 266	535 x 272
Metagenome	529 x 15,132	526 x 56,214	529 x 144	526 x 174

3 CLASSIFICATION

Classification was performed on 12 preparations of the datasets by four classification algorithms. The datasets were the raw data, standardized data and PCA projection versions of the Metagenomic and Metatranscriptomic datasets, with feature reduction by abundance percentile or control/disease variance. The classification algorithms used were Naïve Bayes (NB), Linear Support Vector Machine Classifier (SVC), Random Forest (RF), and K-Nearest Neighbors (KNN). These algorithms were chosen for their popularity, which is related to their relative simplicity and interpretability. Other methods, such as deep neural networks would likely produce more accurate results, but they are much more difficult to interpret and take longer to train. The four algorithms selected work quickly and can provide insight as to the most important features in the dataset, though this was not the focus of this project. All four algorithms were used in their default implementation in the SciKit-Learn Python package.

The NB (Maron, 1961) algorithm works by calculating the probability of each class and the conditional probability of each attribute by each class, based on the frequency of the classes and attributes in the training data. It then uses these probabilities to calculate the probability of a given instance belonging to each class based on its attributes and assigns the classification with the greatest probability. This method is extremely fast, but it also assumes a Gaussian distribution in the data, as well as the independence of all the features. This is a bad assumption in a biological context, as many protein families occur along the same gene pathways, so NB was not expected to produce accurate classifications.

A support vector machine (Ben-Hur, 2001) is an algorithm that seeks to find the best linear discriminant for a dataset. While many possible lines or hyperplanes could separate classes in a dataset, the SVC finds the nearest examples of different classes, called support vectors, and places the discriminant between them. Classification is then assigned based on the position of a new example relative to the discriminant. A major advantage of the SVC is that it is very fast, since the training data can essentially be reduced to the support vectors alone. It is widely used in a biological context because its use of the support vectors makes it resistant to and helps it avoid overfitting on the training data.

A random forest (Ho, 1995) is a collection of decision trees generated from randomly selected features of a training set. A decision tree grows by splitting the data on the fewest features needed to explain the variety dataset and can be followed by new examples for classification. Decision trees are fast and easy to understand, but they are prone to overfitting. An RF addresses this disadvantage by growing many decision trees (often thousands) on random features, and simply having the trees vote on a classification. The voting process helps the RF to deal with noise and overfitting, and it has been shown to be effective in a biological context in the past.

The KNN (Altman, 1992) algorithm is the only unsupervised algorithm that was used for this project. Instead of learning the training dataset first, it just computes the distances from each test item to all the training items. It then orders those distances and chooses the k lowest. The algorithm then assigns the most common class among those k examples to the test item and proceeds to the next one. This algorithm was included because it had been reported to perform poorly in OTU-classification, and a comparison to the supervised techniques was needed.

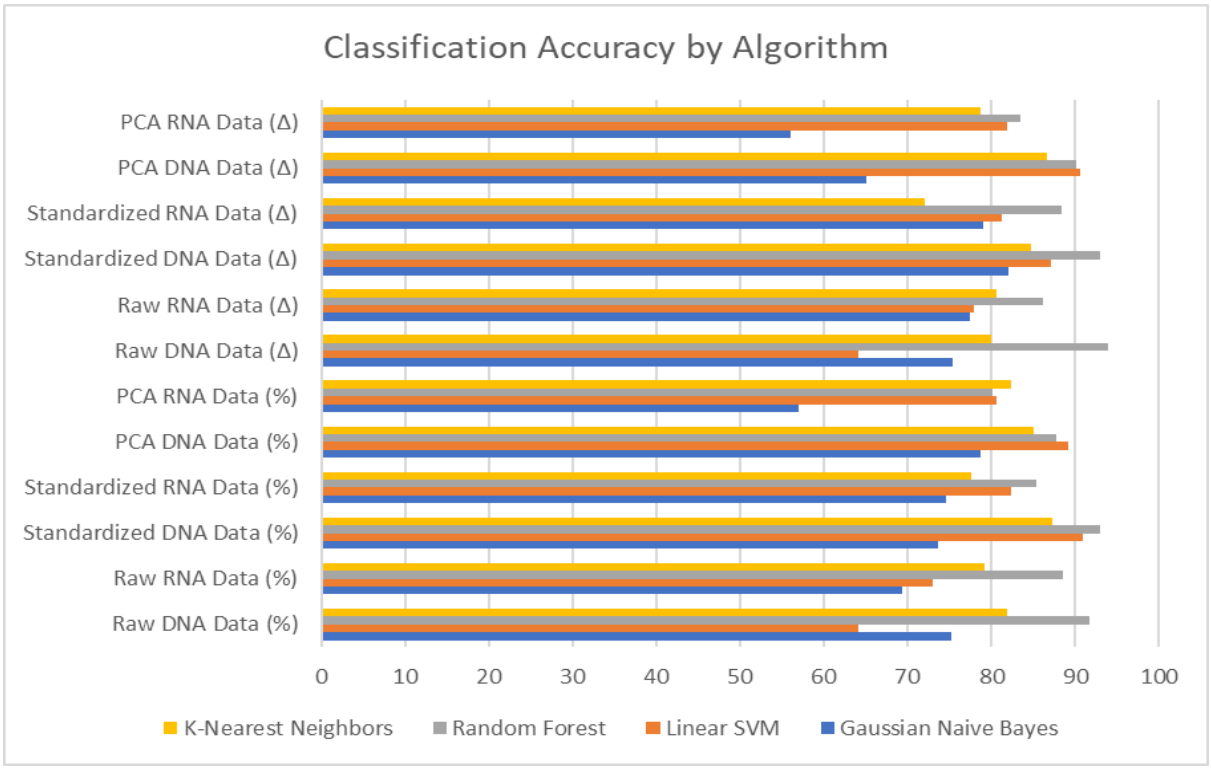
Each algorithm was run on each dataset with a randomly split training set consisting of 80% of the total dataset and stratified by disease state, five separate times. The remaining 20% of the dataset was used as the test set and was classified by the resulting model. The classification accuracy was collected and averaged for each algorithm on each dataset

4 RESULTS

Of the two feature reduction schemes, the Δ Variance method had slightly better performance at 80.68% average accuracy than percentile filtration at 80.38%, though the difference is not significant. Unsurprisingly, the RF model achieved the best classification accuracy, while NB performed the worst. Interestingly, the KNN algorithm performed second-best, despite it having been noted as a generally poor choice in the literature. The performance of the algorithms varied based on the preparation of the data, but in general, the metagenomic data produced higher classification accuracies than the metatranscriptomic data. The only significant exceptions to this pattern were seen in the raw data. In a few cases, the

Table 3: Classification Accuracy

99% Percentile	Gaussian Naive Bayes	Linear SVC	Random Forest	K-Nearest Neighbors	Average of Averages
Raw					
Metagenomic	75.284	64.15	91.7	81.89	**78.256**
Metatranscriptomic	69.346	73.084	88.6	79.252	77.5705
Standardized Data					
Metagenomic	73.586	90.944	93.02	87.358	**86.227**
Metatranscriptomic	74.58	82.43	85.46	77.57	80.01
Principle Components					
Metagenomic	78.68	89.246	87.736	85.096	**85.1895**
Metatranscriptomic	57.01	80.56	80.186	82.43	75.0465
Greatest Δ					
Raw					
Metagenomic	75.434	64.15	93.962	80.002	78.387
Metatranscriptomic	77.432	77.946	86.168	80.562	**80.527**
Standardized Data					
Metagenomic	82.076	87.17	93.02	84.718	**86.746**
Metatranscriptomic	79.064	81.308	88.41	72.148	80.2325
Principle Components					
Metagenomic	65.094	90.568	90.19	86.604	**83.114**
Metatranscriptomic	56.074	81.868	83.552	78.692	75.0465
Overall Average Accuracy	71.97166667	80.28533333	88.50033333	81.36016667	



NB and SVC models achieved higher accuracies, but the greater trend was that metagenomic data was easier to classify correctly. The details are shown in Table 3 and the graphs below.

5 DISCUSSION

The apparent advantage of metagenomic over metatranscriptomic gene family composition data for classification of Crohn's Disease is interesting. The difference in classification accuracy was highly dependent on the algorithm and the preparation of the data and ranged from an insignificant few tenths to a full ten percent. This implies that Crohn's Disease is more dependent on the reservoir of potential protein expression in the gut microbiota than on the most common behaviors of the population. It is also interesting to note that previously reported, OUT-based classification accuracies are typically around 70%. With proper tuning and feature selection, function-based classification could be even more accurate. However, this is not a conclusive analysis one way or the other for several reasons.

First, the samples in general, and the metatranscriptomic samples in particular, were not as controlled as they could have been. They come from individuals that vary widely in demographic details, as well as the duration, severity and treatment strategy for Crohn's Disease. RNA expression is especially sensitive to changes in the environment, such as the presence of antibiotics, the time since the last meal, or even the composition of that meal. A dataset controlling for all of these factors would produce much more useful conclusions. Unfortunately, as extensive as the IBDMDB dataset is, it is not large enough to support this type of analysis while controlling for demographic features, and it's time-related metadata was limited.

Additionally, none of the algorithms used for classification were tuned. Since the only feature of interest was the difference in classification accuracy based on 'omic type, it is possible that many of the classification techniques could perform better with more tuning.

Feature selection could also be improved. For this project the feature reduction strategy was unsophisticated and aimed more at managing the size of the data files and the speed of the analyses. Crohn's Disease, like many other medical conditions, has a complex etiology and is known to have few highly significant factors. This makes diseases with complex etiologies resistant to efficient feature prioritization techniques like PCA. More rigorous feature selection such as stepwise or backward feature selection would be more appropriate, but with 1.64 x 10^6 gene families between these two datasets, such an approach would be extremely resource intensive.

Low-abundance and low-occurrence gene families were also cut from the dataset for the sake of efficiency. It would be interesting to see if any of the low-abundance gene families improve classification accuracy, or if number of gene families itself is correlated with a diagnosis of Crohn's Disease. It seems unlikely that rare features would be associated with disease development considering the size of the gut microbiota population. On the other hand, it is certainly possible for a low abundance gene family to be significant considering how sensitive the microbiota is to equilibrium perturbations.

Finally, it is important to note that translating proteins is not the only function of the transcriptome. Public data on untranslated RNA in the metatranscriptome was unavailable in IBDMDB datasets, though it would be possible to include ncRNA abundance in the same way as a protein family. When such data becomes available, along with better temporal metadata, it is possible that more accurate classification features will be included with them.

REFERENCES

Altman, N. (1992). An Introduction to Kernel and Nearest-Neighbor Non-parametric Regression. *The American Statistician*, 46(3), 175-185. doi:10.2307/2685209

Armour, C. R., Nayfach, S., Pollard, K. S., & Sharpton, T. J. (2019). A Metagenomic Meta-analysis Reveals Functional Signatures of Health and Disease in the Human Gut Microbiome. *MSystems*, *4*(4), 1–15. https://doi.org/10.1128/msystems.00332-18

Asgari, E., Garakani, K., McHardy, A. C., & Mofrad, M. R. K. (2018). MicroPheno: Predicting environments and host phenotypes from 16S rRNA gene sequencing using a k-mer based representation of shallow sub-samples. *Bioinformatics*, *34*(13), i32–i42. https://doi.org/10.1093/bioinformatics/bty296

Asif, M., Martiniano, H. F. M. C. M., Vicente, A. M., & Couto, F. M. (2018). Identifying disease genes using machine learning and gene functional similarities, assessed through Gene Ontology. *PLoS ONE*, *13*(12), 1–15. https://doi.org/10.1371/journal.pone.0208626

Bang, S., Yoo, D. A., Kim, S. J., Jhang, S., Cho, S., & Kim, H. (2019). Establishment and evaluation of prediction model for multiple disease classification based on gut microbial data. *Scientific Reports*, *9*(1), 1–9. https://doi.org/10.1038/s41598-019-46249-x

Ben-Hur, Asa; Horn, David; Siegelmann, Hava; Vapnik, Vladimir N. (2001) Support vector clustering, *Journal of Machine Learning Research*. 2: 125–137.

Ho, Tin Kam (1995). Random Decision Forests *Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal,* QC, 14–16 August 1995. pp. 278–282

Li, F., Hitch, T. C. A., Chen, Y., Creevey, C. J., & Guan, L. L. (2019). Comparative metagenomic and metatranscriptomic analyses reveal the breed effect on the rumen microbiome and its associations with feed efficiency in beef cattle 06 Biological Sciences 0604 Genetics 06 Biological Sciences 0605 Microbiology. *Microbiome*, *7*(1), 1–21. https://doi.org/10.1186/s40168-019-0618-5

Maron, M. E. (1961) Automatic Indexing: An Experimental Inquiry *Journal of the ACM*. **8** (3): 404-417. doi:10.1145/321075.321084

Pasolli, E., Truong, D. T., Malik, F., Waldron, L., & Segata, N. (2016). Machine Learning Meta-analysis of Large Metagenomic Datasets: Tools and Biological Insights. *PLoS Computational Biology*, *12*(7), 1–26. https://doi.org/10.1371/journal.pcbi.1004977

Pearson, K. (1901) On Lines and Planes of Closest Fit to Systems of Points in Space, *Philosophical Magazine*. **2** (11): 559–572. doi:10.1080/14786440109462720.

Qu, K., Guo, F., Liu, X., Lin, Y., & Zou, Q. (2019). Application of machine learning in microbiology. *Frontiers in Microbiology*, *10*(APR), 1–10. https://doi.org/10.3389/fmicb.2019.00827

Vivarelli, S., Salemi, R., Candido, S., Falzone, L., Santagati, M., Stefani, S., … Libra, M. (2019). Gut microbiota and cancer: From pathogenesis to therapy. *Cancers*, *11*(1), 1–26. https://doi.org/10.3390/cancers11010038

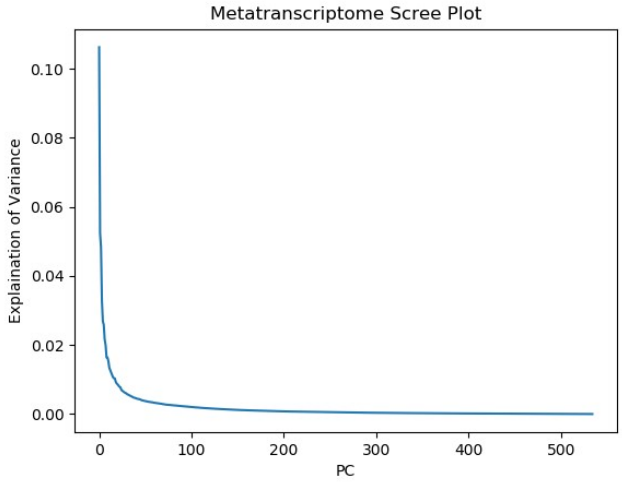
Zuo, T., & Ng, S. C. (2018). The Gut Microbiota in the Pathogenesis and Therapeutics of Inflammatory Bowel Disease. *Frontiers in microbiology*, 9, 2247. doi:10.3389/fmicb.2018.02247

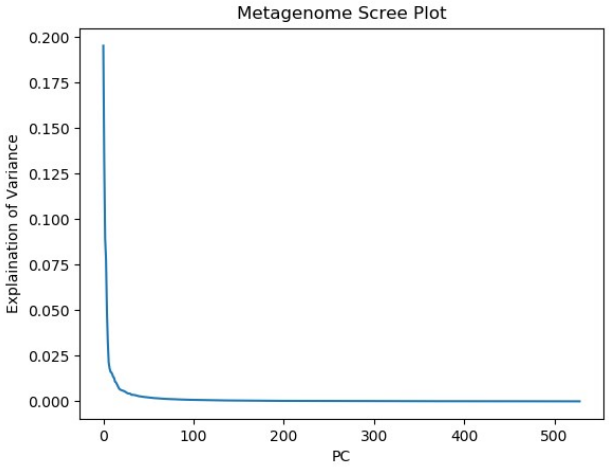
Zhou, Y. H., & Gallins, P. (2019). A review and tutorial of machine learning methods for microbiome host trait prediction. *Frontiers in Genetics*, *10*(JUN), 1–14. https://doi.org/10.3389/fgene.2019.00579

Appendix:

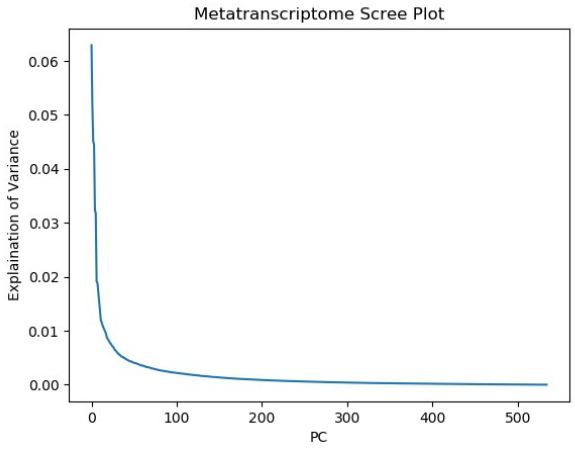
Additional Charts

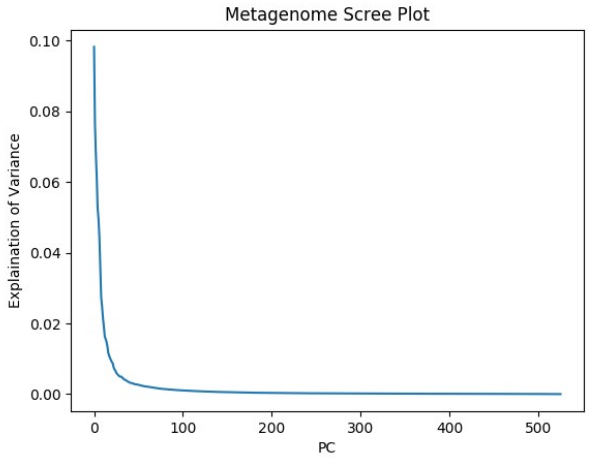
99th Percentile Filtered Scree Plots:



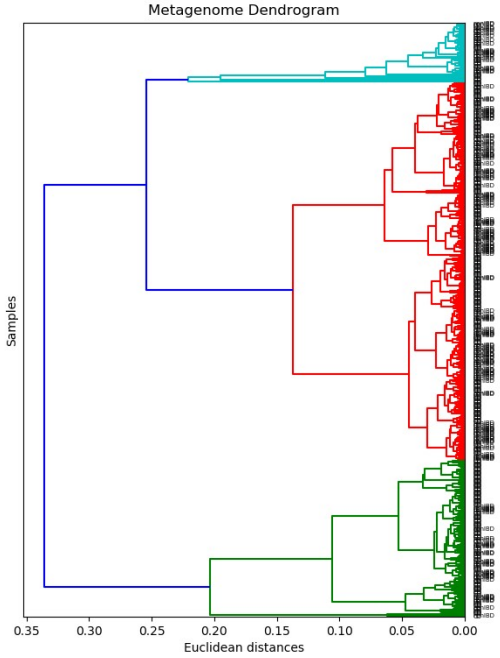


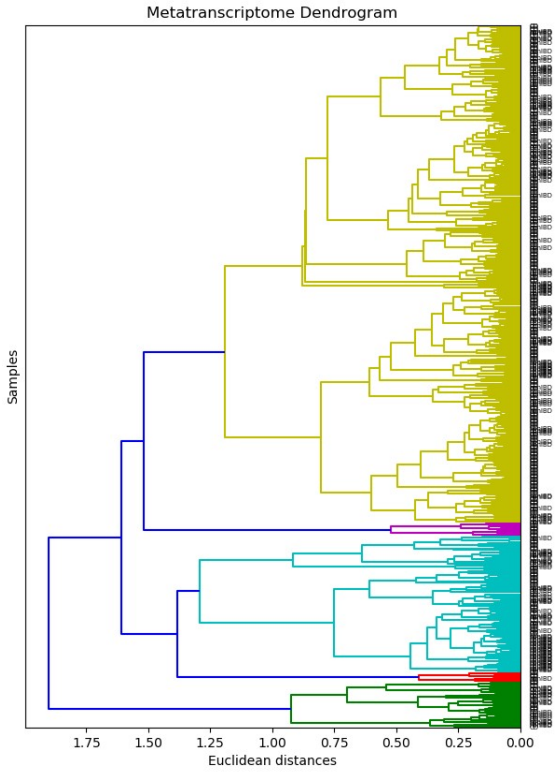
Δ Variance Scree Plots:



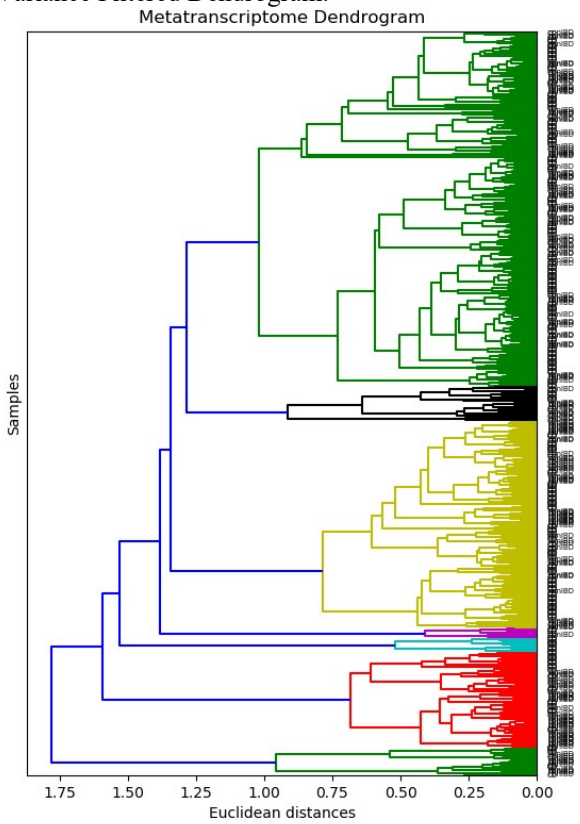


Percentile Filtered Dendrograms:





Variance Filtered Dendrogram:



Code Notes

Code Notes: Data Processing and Preparation
Most commands were executed on a local Windows desktop, the larger files were processed on a Google Virtual Machine.

```
# retrieving URLs
py data_prep.py -s True -u True -d '..\Data\hmp2_metadata_relevant.csv' -o
'D:\Documents\MetaGenomics_Class_Data\Final'
# downloading data
py data_prep.py -g True -d '..\Data\hmp2_metadata_relevant_url.csv' -o
'D:\Documents\MetaGenomics_Class_Data\Final\'
# extracting gene family files
py data_prep.py -e True -d '..\Data\hmp2_metadata_relevant_url.csv' -o
'D:\Documents\MetaGenomics_Class_Data\Final\'

# collecting feature information
py feature_tests.py -d ..\Data\rna_control.csv -f
D:\Documents\MetaGenomics_Class_Data\Final\GeneFamilies\RNA\ -o
rna_control_features
py feature_tests.py -d ..\Data\rna_CD.csv -f
D:\Documents\MetaGenomics_Class_Data\Final\GeneFamilies\RNA\ -o
rna_CD_features
py feature_tests.py -d ..\Data\dna_control.csv -f
D:\Documents\MetaGenomics_Class_Data\Final\GeneFamilies\DNA\ -o
dna_control_features
py feature_tests.py -d ..\Data\dna_CD.csv -f
D:\Documents\MetaGenomics_Class_Data\Final\GeneFamilies\DNA\ -o
rna_CD_features
py feature_tests.py -d ..\Data\hmp2_metadata_dna.csv -f
D:\Documents\MetaGenomics_Class_Data\Final\GeneFamilies\DNA\ -o
dna_full_features
py feature_tests.py -d ..\Data\hmp2_metadata_rna.csv -f
D:\Documents\MetaGenomics_Class_Data\Final\GeneFamilies\RNA\ -o
rna_full_features
# saving features in 99th percentile for full datasets
py feature_tests.py -d .\rna_full_features.csv -t True -o
rna_full_trimmed.data
py feature_tests.py -d .\dna_full_features.csv -t True -o
dna_full_trimmed.data

# trimming features based on difference between control and disease groups
# values < 1.5 std or > 2.5 std from the mean are retained
py feature_tests.py -d .\rna_control_features.csv -d2 .\rna_CD_features.csv
-o rna_diff_attr2.data
py feature_tests.py -d .\dna_control_features.csv -d2 .\dna_CD_features.csv
-o dna_diff_attr2.data
# these features are too numerous to examine

# assembling the metatranscriptomic total family composition dataset
# rna data, percentile trimmed
py data_prep.py -x 'rna_tot' -f
'D:\Documents\MetaGenomics_Class_Data\Final\GeneFamilies\RNA\' -d
'..\Data\hmp2_metadata_relevant_url.csv' -o '..\Data\' -p
rna_full_trimmed.data
```

```
# difference between groups
py data_prep.py -x 'rna_tot' -f
'D:\Documents\MetaGenomics_Class_Data\Final\GeneFamilies\RNA\' -d
'..\Data\hmp2_metadata_relevant_url.csv' -o '..\Data\' -p rna_diff_attr2.data

# randomly sampling DNA dataset to match size of RNA set
py .\feature_tests.py -s True -d '..\Data\dna_control.csv' -d2
..\Data\dna_CD.csv  -o '..\Data\hmp2_metadata_reduced_dna.csv'

# assembling the metagenomic total family composition dataset
# dna data, percentile trimmed
py data_prep.py -x 'dna_tot' -f
'D:\Documents\MetaGenomics_Class_Data\Final\GeneFamilies\DNA\' -d
'..\Data\hmp2_metadata_reduced_dna.csv'  -o '..\Data\' -p
dna_full_trimmed.data
# difference between groups
python3 data_prep.py -x 'dna_tot' -f ../data/Sampled/ -d
../hmp2_metadata_reduced_dna.csv  -o ../datasets/  -p ./dna_diff_arrt2.data

# generating dendrograms
py data_process.py -d ..\Data\Datasets\rna_tot_df.csv -o
..\Data\Plots\RNA_dendro.png -ds RNA -hc True
py data_process.py -d ..\Data\Datasets\dna_tot_df.csv -o
..\Data\Plots\DNA_dendro.png -ds DNA -hc True
py data_process.py -d ..\Data\Datasets\rna_tot_diff_df.csv -o
..\Data\Plots\RNA_diff_dendro.png -ds RNA -hc True
python3 data_process.py -d ../datasets/dna_diff_df.csv -o ../DNA_dendro.png -
ds DNA -hc True
# did not work, memory error on local machine, no output on Google VM

# plotting gene family abundance when trimmed by percentile
py data_process.py -d ../Data/Datasets/rna_tot_df.csv -o na -d2
..\Data\Datasets\dna_tot_df.csv -f True -o
..\Data\Plots\fam_abund_percentile.png
python3 data_process.py -d ../datasets/rna_diff_df.csv -o na -d2
../datasets/rna_diff_df.csv -f True -o ../fam_abund_diff.png

# standardizing datasets
py data_process.py -d ../Data/Datasets/rna_tot_df.csv -o
../Data/Datasets/RNA_std.csv -s True
py data_process.py -d ../Data/Datasets/dna_tot_df.csv -o
../Data/Datasets/DNA_std.csv -s True
py data_process.py -d ../Data/Datasets/rna_tot_diff_df.csv -o
../Data/Datasets/RNA_diff_std.csv -s True
python3 data_process.py -d ../datasets/dna_diff_df.csv -o
../datasets/DNA_diff_std.csv -s True

# generating scree plots
# percentile
py data_process.py -d ..\Data\Datasets\RNA_std.csv -o
..\Data\Plots\RNA_scree.png -ds RNA -ps True
py data_process.py -d ..\Data\Datasets\DNA_std.csv -o
..\Data\Plots\DNA_scree.png -ds DNA -ps True
```

```
# variance
py data_process.py -d ..\Data\Datasets\Difference_Filtered\RNA_diff_std.csv -
o ..\Data\Plots\RNA_scree.png -ds RNA -ps True
python3 data_process.py -d ../datasets/DNA_diff_std.csv -o ../DNA_scree.png -
ds DNA -ps True
******download

# applying 95% PCA to both datasets
py data_process.py -d ..\Data\Datasets\RNA_std.csv -o
..\Data\Datasets\rna_pca.csv -pca True
py data_process.py -d ..\Data\Datasets\DNA_std.csv -o
..\Data\Datasets\dna_pca.csv -pca True
python3 data_process.py -d ../datasets/DNA_diff_std.csv -o
../datasets/dna_diff_pca.csv -pca True
py data_process.py -d ..\Data\Datasets\RNA_std.csv -o
..\Data\Datasets\rna_diff_pca.csv -pca True

# formatting dfs
# percentile data
py .\data_process.py -d
..\Data\Datasets\Percentile_Filtered\dna_filter_df.csv -o na -c True;
py .\data_process.py -d
..\Data\Datasets\Percentile_Filtered\rna_filter_df.csv -o na -c True;
py .\data_process.py -d
..\Data\Datasets\Percentile_Filtered\dna_filter_pca.csv -o na -c True;
py .\data_process.py -d
..\Data\Datasets\Percentile_Filtered\rna_filter_pca.csv -o na -c True;
py .\data_process.py -d
..\Data\Datasets\Percentile_Filtered\RNA_filter_std.csv -o na -c True;
py .\data_process.py -d
..\Data\Datasets\Percentile_Filtered\DNA_filter_std.csv -o na -c True

# difference data
python3 data_process.py -d ../datasets/dna_diff_df.csv -o na -c True;
py .\data_process.py -d ..\Data\Datasets\Difference_Filtered\rna_diff_df.csv
-o na -c True;
py .\data_process.py -d ..\Data\Datasets\Difference_Filtered\dna_diff_pca.csv
-o na -c True;
py .\data_process.py -d ..\Data\Datasets\Difference_Filtered\rna_diff_pca.csv
-o na -c True;
py .\data_process.py -d ..\Data\Datasets\Difference_Filtered\RNA_diff_std.csv
-o na -c True;
python3 data_process.py -d ../datasets/DNA_diff_std.csv -o na -c True

# CSVs converted to lighter pickle files, for example:
py .\data_process.py -d ..\Data\Datasets\Difference_Filtered\DNA_diff_std.csv
-o na -c True

# Classifications performed with classifiers.py by passing the file and the
model as parameters, for example:
py .\classifiers.py -d ..\Data\Datasets\Difference_Filtered\rna_diff_df.pkl -
m SVC

BUGS: Small bug in the dataset concatenation script that resulted in a few
duplicate individuals being included. However, there were no duplicate rows,
meaning that these samples were taken at different times from the same indi-
vidual.
```