CS4220 Pathogen Detection Project Report
Choo Min Han (A0199125X), Lim Yee Chern (A0216606J)

Introduction

Phylogenetic studies estimate that microbial life has coexisted with wild hominids and evolved with their hosts [1]. While symbiosis between some microbes and their hosts suggests co-adaptation for mutual benefit, others appear to cause disease. Many microbes are impossible to cultivate in laboratory media so mechanistic effects are difficult to characterise [2]. While model organisms can be used to explore disease causation and pathogenesis, these models only approximate some human diseases and do not necessarily represent true interactions between these microbes and a human host [3]. Coupled with machine learning approaches and increasing accessibility of Next-Gen Sequencing, metagenomic analysis has been proven as a viable means to identify known and novel pathogens. Using a k-mer based approach avoids problems with sequence-based approaches such as alignment and can utilise unannotated DNA. Using a machine learning approach, we explored using classifiers to accurately detect the presence of 10 different pathogens from a complex metagenome, given the 6-mer profile of the sequenced reads. The 6-mer frequency profile of each read is kept as its features. Among various models, we found an ensemble-based Random Forest (RF) classifier with decision trees splitting by Gini coefficient had the best performance.

Methods

Data Exploration

The training and validation dataset primarily consisted of decoys. After sampling a subset of the data provided, Uniform Manifold Approximation and Projection (UMAP) was used to visualise the data as a 2D projection of all features. We found that reads from each pathogen formed tight clusters, whereas decoy reads formed multiple clusters - one was larger but equally dense, but a large number were found co-clustered with reads from other pathogens (**Figure A1**). This is representative of metagenomic data from a microbiome – related organisms can share highly identical sequences that have similar k-mer profiles, and thus reads from these organisms can appear to cluster together. Also, low pathogen abundance results in fewer sequenced reads from the pathogen relative to the total DNA that can be sequenced from a sample. The fraction of decoy reads subsampled for training a classifier should be much higher in proportion to the fraction of pathogenic reads to ensure model accuracy with realistic test data. Informally, a model trained with more decoy reads can better distinguish between the decoys and pathogenic reads. Subsampling with more decoys will ensure that the classifiers are trained to identify decoy reads that share similar features to true positive reads.

As the number of k-mers is exponentially related to the length of k-mers chosen, a full 6-mer profile would be highly dimensional and would require preprocessing before it can be used for training a model. Prokaryotic DNA is double-stranded, so sequences and their reverse complements are equally abundant, so k-mer profile analysis can be performed on data with canonical k-mers as features. This was the case in the data, reducing the number of features from 4096 6-mers to 2080. Highly dimensional data is prone to model overfitting and expensive model training, especially with the cardinality of the dataset. Additional steps to reduce the dimensionality of the data were sought. Further analysis through Principal Component Analysis (PCA) and chi-square feature selection were performed to provide greater dimensional reduction during model training.

Classifier performance was determined by the average precision score for the classifier on the validation datasets. Ideally, a classifier should be scored high when predicting a pathogen correctly but penalised when falsely identifying pathogens not present. Thus a measure of precision is sufficient as a measure of classifier performance.

As a naive approach, we built untuned classifiers from scikit-learn python package and continued tuning the parameters of the classifiers with the best precision. The training dataset was split with a 70/30 train/test ratio for these models Classifiers using K Nearest Neighbours (kNN) and Random Forest (RF) approaches presented better precision and were chosen for further tuning (**Table A1**).

kNN

The clustering of data from the UMAP seems to suggest good support for a kNN classifier. While clusters are easy to define visually, arithmetic determination of clusters for data with high dimensionality and cardinality is computationally expensive, resulting in lengthy prediction times. PCA was applied to reduce the dimensionality of the data and reduce feature redundancy. The cumulative explained variance of each principle component in the new feature space was measured against the number of principle components (**Figure A2**). Multiple kNN classifiers were trained by fitting to the train set with a feature space with reduced dimensions to determine a good PCA transformation that could preserve the performance of the kNN classifier.

While initial exploration using the train/test split of the original training dataset with kNN models showed promising results, this performance did not carry over to testing with the validation sets provided. (**Table A1).** Also, we found that a reduced feature space that preserved >95% of the variance in the data had poor performance when used to predict on the validation set (**Table A2**). To improve on the performance, we attempted to normalise the data before PCA, but this did not yield any significant improvement on the validation set. We suspect that clusters were sparse as there were a large number of features remaining, and offer our explanation in the Conclusion section. While a majority of the predictions had non-zero precision, the classifiers had poor performance due to a large number of false positives, which we suggest could be caused by the decoys being found in clusters of reads from the various pathogens. By increasing the number of neighbours compared in the model, the classifier only found slightly improved performance.

Random Forest

Random Forest (RF) classifiers are ensemble-based classifiers that utilise multiple randomly generated decision trees. While single decision trees are easier to interpret, they are prone to overfitting and thus require careful pruning to ensure accurate prediction of performance. RF classifiers employ bootstrapping and randomised subspace sampling to build decision trees and return an aggregated result of the decision trees, which reduces the risk of overfitting at the cost of being more difficult to interpret. Each node of a decision tree uses a random subset of features to classify reads (i.e. random subspace sampling), and each decision tree is trained using a random subsample of the data drawn with replacement (i.e. bootstrapping). The overhead of training RF classifiers is still costly, as each decision tree is trained independently, but queries are quickly processed. We decided to train an RF classifier that utilises decision trees with node splitting scored using the Gini coefficient. The classifier then predicts a set of probabilities for the presence of each pathogen per decision tree and returns the aggregated probability for each pathogen.

Chi-squared test was performed to rank features by their correlation to the class labels. Then, a suitable set of features was selected for training the RF classifier. Based on the performance of the models, the top 900 features were sufficient for comparable precision and would provide the greatest dimensional reduction (**Figure A3**). Finally, the classifiers were tuned for an appropriate threshold. Briefly, the threshold represents the minimum confidence required to determine if a read originated from any pathogen. A threshold that is overly stringent will have a low precision as the classifier may not be sufficiently confident on some true positives, but a weak threshold will suffer from a large number of false positives that were not supported with good confidence.

Results

RF classifier was built using 200 decision trees with sample bootstrapping. Changing the fraction of decoy reads in training data affected the accuracy of the classifier. To determine an optimal fraction, training datasets were subsampled with different ratios of decoy reads to pathogenic reads. We found that using 5x decoys per pathogenic read to train the classifier resulted in the best performance when tested on the validation dataset (**Figure A4**), most likely due to the fraction of decoy reads that were present in the dataset (**Table A3**). Further improvement to the performance of the model is possible when training with a larger fraction of decoy reads, but memory constraints did not permit the construction of an RF classifier with a training dataset larger than approximately 150,000 reads. Finally, the probability threshold was adjusted for the classifier. Our model performed best when only filtering out predictions with an aggregated probability lower than 0.55 (**Figure A5**), with perfect precision on 14 out of 16 validation datasets, and an average precision of 0.95 (**Table A4).**

Conclusion

kNN classifier performed poorly despite model tuning and increasing the number of neighbours used for class prediction. Data with a large number of features cannot be easily classified using kNN. The time complexity of prediction using kNN is O(k*n*d) for each test instance, where k is the number of neighbours, n is the number of training instances, and d is the dimensionality of the data. In practice, a test dataset can contain hundreds of thousands of instances which result in lengthy prediction times. Additionally, kNN, and other methods which utilise pairwise comparison, suffer from the curse of dimensionality. Briefly, as the number of dimensions increases linearly, the distance between two points increases exponentially [4]. As a result, high dimensional data is sparse and the quality of clusters is poor, so features become poorly correlated with the classes.

RF Classifier exhibited the most promising results based on the validation set. Due to the large number of features even after feature selection (necessary to preserve model performance), one may obtain better classifiers in random subspaces than in the original feature space. The combined decision of many such classifiers may be superior to a single classifier constructed on the original training set in the complete feature space, given that this method of feature reduction preserves the euclidean geometry of the data [5]. Accuracy of RF classifiers can potentially be improved by using more decision trees, or exploring other ensemble methods that utilise SVM or kNN with random subspace for data splitting.

On the test dataset, the 900 best features are extracted and the same RFC model is used to predict the pathogens present. The predictions are recorded in **Table A5.**

References

1. Ochman, H. et al. Evolutionary relationships of wild hominids recapitulated by gut microbial communities. *PLoS Biol.* 8, e1000546 (2010).
2. Baumann, P. & Moran, N. A. Non-cultivable microorganisms from symbiotic associations of insects and other hosts. *Antonie van Leeuwenhoek* 72, 39–48 (1997).
3. Greenblum, S., Turnbaugh, P. J. & Borenstein, E. Metagenomic systems biology of the human gut microbiome reveals topological shifts associated with obesity and inflammatory bowel disease. *Proc. Natl Acad. Sci. USA* 109, 594–599 (2012)
4. Radovanović, Miloš; Nanopoulos, Alexandros; Ivanović, Mirjana. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*. 11: 2487–2531. (2010)
5. Lim, Nick and Robert J. Durrant. Linear dimensionality reduction in linear time: Johnson-Lindenstrauss-type guarantees for random subspace. *arXiv: Machine Learning* (2017)

<u>**Appendix**</u>

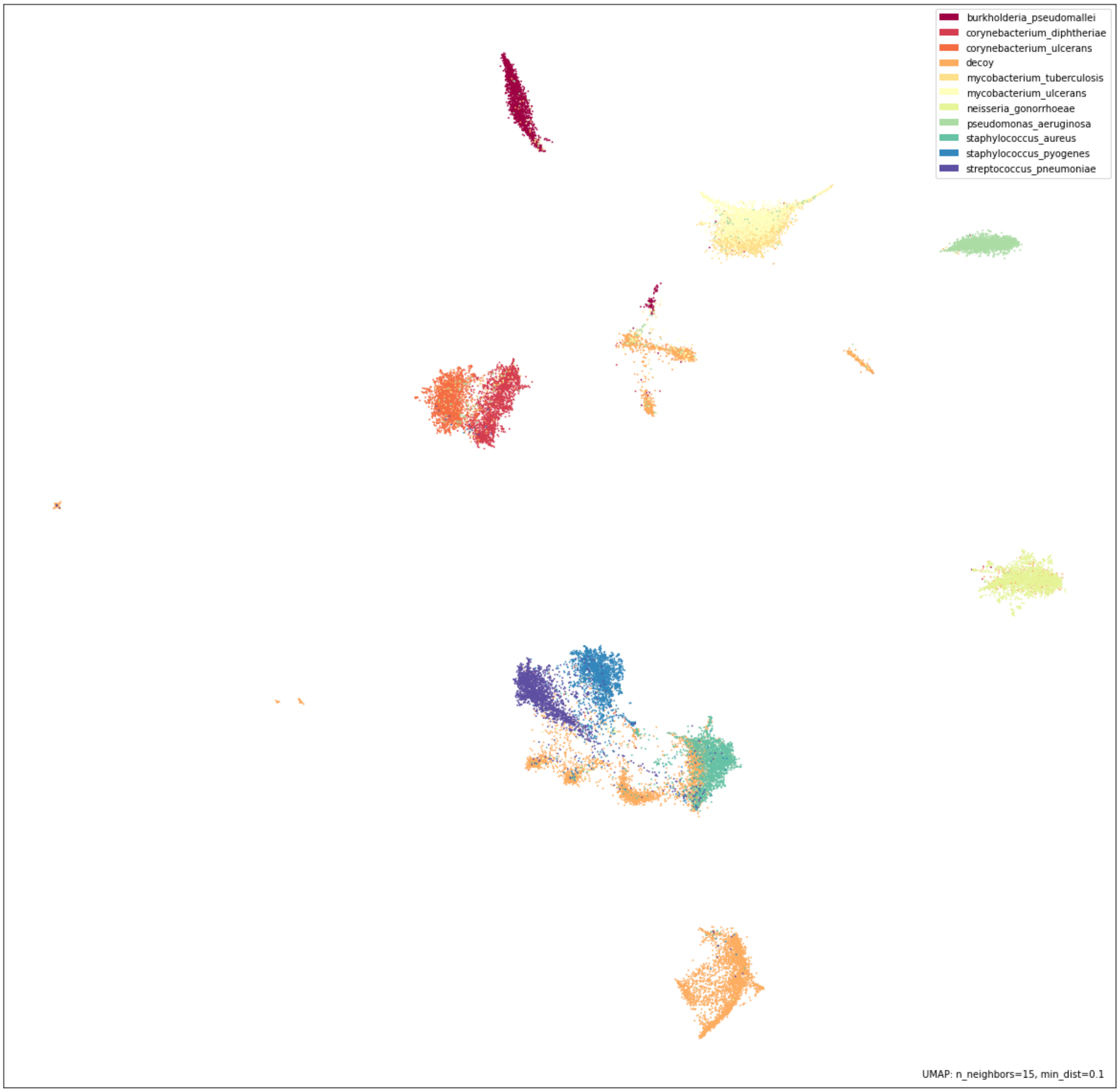

Figure A1. *2D Projection of the data subsample used to train Random Forest Classifier. 10,000 decoy reads and 2078 reads from each of 10 pathogens were drawn without replacement.*

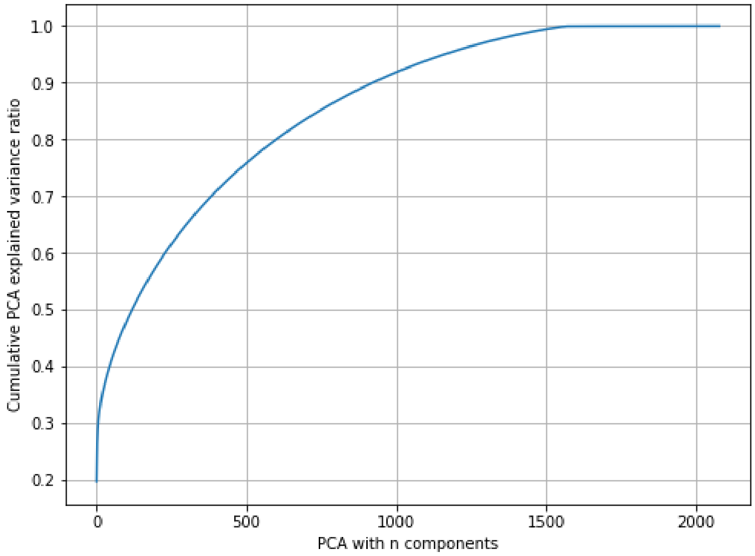

***Figure A2**. Cumulative Explained Variance of data for n-best PCA components. Notably, 0.90 explained variance can be achieved with 922 principle components, and 0.95 explained variance with 1622 principle components.*

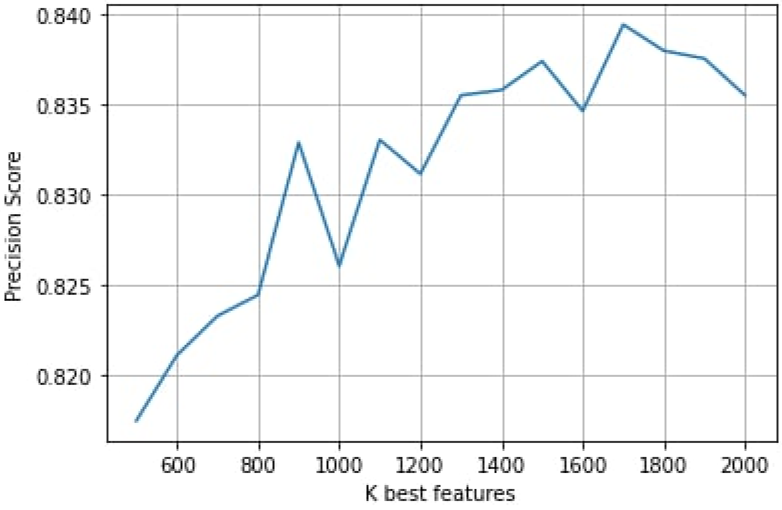

***Figure A3**. Random Forest classifiers trained with variable k-best features (ranked by chi-square test) found relatively similar average precision scores between 0.80 and 0.85 when tested with the validation dataset.*

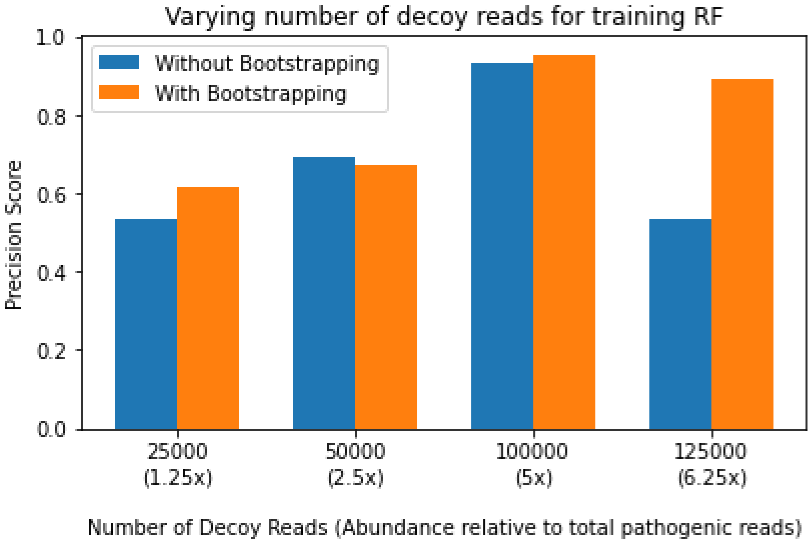

Figure A4. *Random Forest classifiers trained with different fractions of decoy reads had varying average precision scores. The number of reads from the 10 other pathogens were kept constant at 2078 reads per pathogen.*

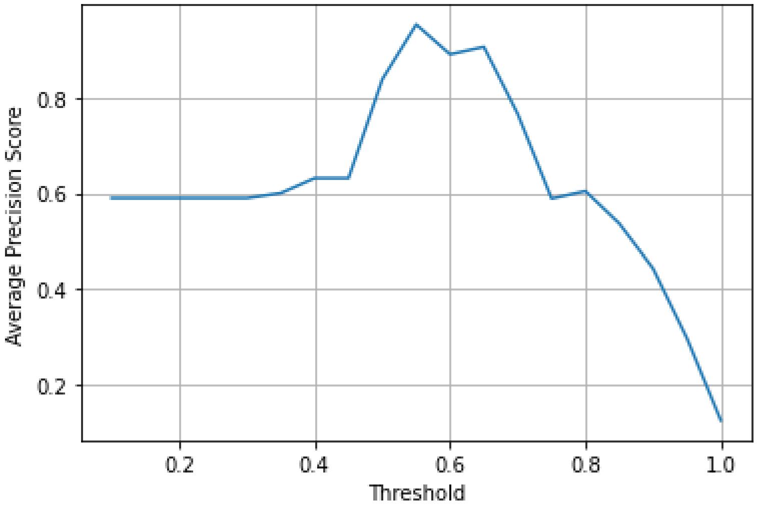

Figure A5. *Random Forest classifier predictions were filtered using different thresholds for probability and scored based on the average precision score when tested with the validation dataset. A threshold of 0.55 gave the greatest average precision score.*

Model		Average Precision
kNN	125	0.366
	150	0.350
	175	0.316
SVM		0.126
LDA		0.126
DT		0.120
RF		0.350

***Table A1**. K Nearest Neighbours (kNN) and Random Forest (RF) Classifiers exhibit higher precision. kNN with 125, 150 and 175 neighbours were found to have similar precision scores.*

Models (PCs)	Average Precision
Default	0.400
PCA60% (286)	0.291
PCA70% (387)	0.394
PCA80% (605)	0.394
PCA90% (928)	0.394
PCA (1500)	0.361

***Table A2**. Varying levels of dimension reduction PCA for kNN and precision on validation dataset. Number of principle components (PCs) used for each model are given in parentheses.*

	Fraction of Decoys	Fraction of All Pathogens	Comparative Abundance (-fold)
Patient 1	0.97995	0.020049	48.88
Patient 2	0.99681	0.0031872	312.75
Patient 3	0.99233	0.0076695	129.39
Patient 4	0.99966	0.00034494	2898.02
Patient 5	0.93117	0.068833	13.53
Patient 6	0.92605	0.073954	12.52
Patient 7	0.99907	0.00093044	1073.76
Patient 8	0.89310	0.10690	8.35
Patient 9	0.99950	0.00050454	1981.00
Patient 10	0.95433	0.045671	20.90

***Table A3**. Comparative abundance of decoy and total pathogen reads in the first 10 validation datasets.*

	Precision
Patient 1	1.0
Patient 2	1.0
Patient 3	1.0
Patient 4	1.0
Patient 5	0.5
Patient 6	1.0
Patient 7	1.0
Patient 8	1.0
Patient 9	1.0
Patient 10	1.0
Patient 11	1.0
Patient 12	1.0
Patient 13	1.0
Patient 14	0.75
Patient 15	1.0
Patient 16	1.0

***Table A4**. Precision score of our final Random Forest classifier on the sixteen validation datasets*

	Predicted pathogens
Patient 1	pseudomonas_aeruginosa
Patient 2	pseudomonas_aeruginosa
Patient 3	corynebacterium_ulcerans pseudomonas_aeruginosa
Patient 4	mycobacterium_tuberculosis mycobacterium_ulcerans staphylococcus_pyogenes
Patient 5	decoy
Patient 6	burkholderia_pseudomallei corynebacterium_diphtheriae mycobacterium_tuberculosis mycobacterium_ulcerans pseudomonas_aeruginosa
Patient 7	decoy

Table A5*. Final prediction on test datasets*