Transactions on Engineering and Computing Sciences - Vol. 13, No. 04
Publication Date: August 25, 2025
DOI:10.14738/tmlai.1304.19266.

Pödör, Z. & Hekfusz, M. (2025). Feature Selection and an Ensemble Framework for Metagenomic Data. *Transactions on Engineering and Computing Sciences*, 13(04). 116-143.

Feature Selection and an Ensemble Framework for Metagenomic Data

Zoltán Pödör
Eötvös Loránd University, Faculty of Informatics,
Budapest, H-1117, Hungary

Máté Hekfusz
Eötvös Loránd University, Faculty of Informatics,
Budapest, H-1117, Hungary

ABSTRACT

Genome data, characterized by its high dimensionality and complexity, presents significant challenges for computational analysis and biological interpretation. Feature selection plays a crucial role in reducing dimensionality, improving model interpretability, and enhancing predictive performance by identifying the most informative genomic attributes. In this study, we construct a robust, generalisable ensemble framework for the feature selection and ML classification of metagenomic data. The framework incorporates six different feature selection algorithms of different types working in an ensemble. We comprehensively assess four ML classifiers to pair with them and three aggregation methods to combine their results, testing numerous configurations to find which ones perform best. Our result shows that Random Forest is a general and reliable algorithm for metagenomic datasests and consistent with the literature, we found that feature selection universally improves classification performance, though this improvement varies per dataset and, on non-wrapper methods, depends on choosing the right subset size. When looking at their best scores, the six FS algorithms performed broadly similarly across the data, with the largest differences being on the hardest-to-classify datasets, where mRMR and Boruta edged out the others.

Keywords: Feature Selection, Classification, Ensemble Framework, Genome Data.

INTRODUCTION

The human body – like all other life on Earth – is home to trillions of microorganisms, like bacteria, viruses, fungi, and so on. Their numbers are so vast that there are in fact more bacterial cells inhabiting the body than human cells [1]. The human microbiome (as these microbes are collectively called) affects our health: disturbances in the balance of the microbiome have been linked to a variety of conditions and diseases: different types of cancer [2,3], diabetes [4], even asthma [5] and obesity [6].

Metagenomics is the field devoted to studying genetic information taken directly from the environment. Unlike classic microbiological methods, which focus on a single, clonally cultured organism at a time, metagenomics analyses whole communities of microbes within their natural environment – such as the human body – giving a deeper and more diverse picture of a microbiome and its effects, though at the cost of not sequencing the whole genome of any particular species [1]. The field truly took shape with the appearance of next-generation sequencing (NGS) technologies, which allow researchers to process most genomes in an environmental sample at high speed and low cost [7]. Today, there are millions of metagenomic samples available in public databases like the Sequence Read Archive [8]. In tandem with this sequencing revolution, researchers have started applying machine learning (ML) methods to process and analyse the huge amounts of data, with promising results: ML models have been shown to be effective in classifying microbial features, characterising state-specific microbial signatures, and most importantly, predicting diseases [9].

In its default state, however, most metagenomic data is a poor fit for most of the machine learning algorithms. One common type of metagenomic data is species abundance, where samples with millions of raw sequence reads are processed into tables which contain the relative abundances of each microbial feature in that sample. The tables for each sample can then be combined to form a frequency table of the whole dataset, with its dimension given by the total number of microbial features (taxonomically identified, down to a genus, species, or even strain level [10]) found in the dataset's samples. This dimension is much higher, often orders of magnitude larger than the number of samples available for analysis [11,12,13]. Known as the 'curse of dimensionality', this mismatch can lead to long ML training times and overfitted predictive models [14]. Metagenomic data is also known to be sparse (some features only appear in a handful of samples; in the rest, its abundance is zero) and noisy (there are many features present that are irrelevant for the task of disease prediction), further reducing the performance of ML models [12]. In order to overcome these problems and make metagenomic data suitable for machine learning, it must be pre-processed: its dimensionality must be significantly reduced, while irrelevant features and noise must be removed. One of the best ways to achieve these is through feature selection (FS), which has become a vital part of dealing with genomic data of all kinds [15].

Feature selection is the process of selecting a subset of the most informative and relevant features from the total feature set. Unlike other dimensionality reduction methods, such as principal component analysis (PCA), feature selection does not alter features or create new ones; it only picks out a subset. This makes it ideal for metagenomics, as researchers are often not just concerned with creating a good predictive model, but also with identifying the specific microbial taxa that play crucial roles in predicting diseases [16]. The goal of feature selection is to find the optimal subset of features such that ML training is quicker, the resulting model is more accurate, and no important information is lost in the process [14].

There have been countless feature selection methods used on genomic, and specifically metagenomic, data and there is determined, that no single FS method is ideal in every case [12,14]. As such, some researchers utilise multiple FS algorithms in tandem, combining their findings in some manner to produce a more optimal feature set – a process known as ensemble feature selection [15]. While ensemble frameworks have been created for other types of genomic data [17,18], there have been relatively few studies that specifically use metagenomic data [19]. Those that do usually consider only a few datasets [12,20], which inherently limits their generalisability. Studies that analyse a wider range of data often only use one type of feature selection [10] or use different dimensionality reduction methods altogether [21,22]. Additionally, the general direction of ML research in metagenomics (as in many other fields) seems to be towards neural network and deep learning-based architectures [21,22,23], which, while promising superior performance, require powerful computational resources that might not be available to researchers or other, non-academic users who could benefit from the disease prediction and biomarker identification capabilities of these frameworks.

In this paper our aim was to construct a robust, generalisable ensemble framework for the feature selection and ML classification of metagenomic data. The framework incorporates six different feature selection algorithms of different types working in an ensemble. We comprehensively assess four ML classifiers to pair with them and three aggregation methods to combine their results, testing numerous configurations to find which ones perform best. We process six open-source metagenomic disease datasets totalling more than a thousand samples through our framework, contributing a wide array of experimental results to the literature. Beyond disease prediction, we also use the framework to identify important biomarkers for each disease. Importantly, every algorithm we use has low computational demand and does not require powerful hardware, making our work accessible to everyone (without using a supercomputer).

METHODS

In this section, first we give a short description of the different categories of feature selection algorithms and describe how they operate, while also reviewing their usage in genomic big data analysis. To conduct a more thorough review, we consider not just metagenomic studies, but studies working with other kinds of genomic data as well. The issues that feature selection is meant to solve (high dimensionality, low sample size, sparse and noisy data) are present in essentially all genomic data [15,16], hence these studies are also relevant to our work. After that there is a list of the applied ML algorithms and the description of the used datasets.

FS-methods

Table 1: Summarization of FS methods

Basic methods	Filter methods
	Wrapper methods
	Embedded methods
Advanced methods	Hybrid methods
	Ensemble methods

Basic Methods

Filter Methods:

Filter methods are independent from the ML classifiers and conduct statistical tests on the features to see how much they correlate with the target classes (Fig.1). Features are then ranked based on the results of these tests; those above a certain threshold are selected as the subset. Filter methods are considered the simplest and least computationally intensive of the three categories [14]. This makes them easily scalable, which is important for the high-dimensional nature of metagenomic data.

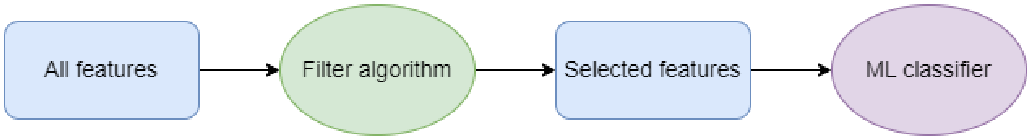

Figure 1: The process of feature selection with filter methods.

Filter methods can be univariate or multivariate. Univariate tests, such as the Chi-squared and Fisher's exact test, are the fastest, but they ignore interactions between features, which in a complex, interconnected microbial community can be detrimental. As such, they were mostly used in older studies, with information gain-based algorithms being their most common form [26]. Multivariate tests are slower, but they do account for some of these interactions. Hacilar et al. [27] used the minimal-redundancy-maximal-relevance (mRMR) multivariate FS method, among others, on an Inflammatory Bowel Disease (IBD) dataset to find the subset of features most associated with the disease. Urbanowitz et al. [28] analysed the popular Relief family of multivariate FS methods on simulated metagenomic datasets, finding that they accurately detected two-way microbial interactions.

Wrapper Methods:

Unlike filters, wrapper methods are invariably tied to a given ML classifier: they use said classifier to evaluate different combinations of features and select the subset that performs the best (Fig.2).

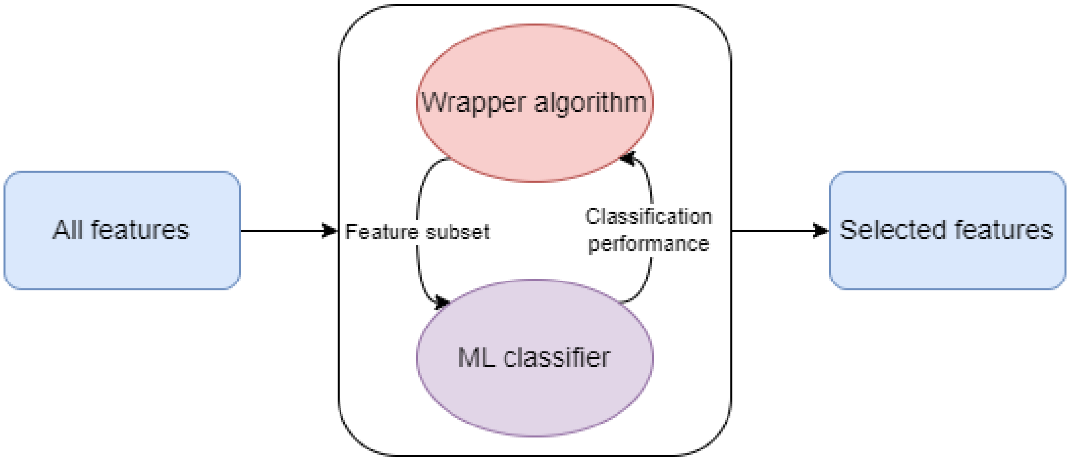

Figure 2: The process of feature selection with wrapper methods.

Because they inherently look for features that perform well with a classifier, wrapper methods produce higher classification accuracy than filters. Their main issue is cost: given the high-dimensional nature of metagenomic data, evaluating every possible subset can be computationally infeasible, requiring search strategies to narrow down the options, Wrappers provide the optimal feature subset for their chosen ML classifier, that subset might not be optimal for other classifiers, which limits the versatility of wrapper methods [14]. Despite their higher cost, wrapper methods have been used extensively in genomics. He et al. [30] devised a novel wrapper FS algorithm based on the mRMR filter to predict genetic traits. Kavakiotis et al. [31] also created a new wrapper, Frequent Item Feature Selection (FIFS), which outperformed other FS methods in the informative marker selection task. Shen et al. [32] used the Boruta wrapper, paired with the Random Forest classifier, to find microbes from the gut microbiome that were important to predicting schizophrenia.

Embedded Methods:

Embedded methods integrate feature selection and ML model training into one step: during training, the ML algorithm automatically determines the importance of each feature (Fig.3). These methods can be considered a middle ground between filters and wrappers: like filters, they are reasonably fast, and like wrappers, they consider the characteristics of the classifier to achieve higher performance [14].

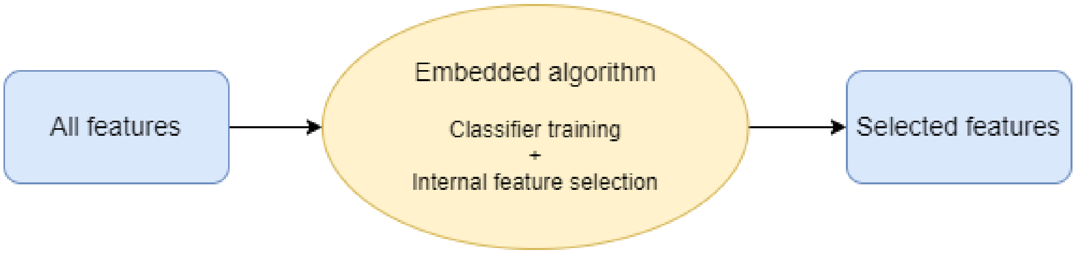

Figure 3: Feature selection with embedded methods.

The two main types of embedded methods are decision tree-based and regularization-based algorithms. Decision tree-based methods (such as Random Forest or Gradient Boosting) are effective at uncovering feature interactions, but since they rank every feature rather than choosing a subset, they do not necessarily eliminate redundant features [14]. Regularization-based methods (LASSO being a common one) are the opposite: by penalizing features that are not relevant to the model, they eliminate redundancy, but they do not implicitly cover feature interactions [33]. Kumar & Rath [34] used Support Vector Machines (SVM) as an embedded way of feature selection, along with statistical filter methods, on leukaemia datasets. Sasikala et al. [35] proposed a new Genetic Algorithm (GA), integrating it with four different ML classifiers to produce a highly accurate model for breast cancer diagnosis.

Advanced Methods

The high dimensionality of genomic (including metagenomic) datasets means that most FS methods are not stable: meaning that the features they select vary severely between similar datasets, or even between subsets sampled from a single dataset [36]. To alleviate this problem, and to create techniques that are more generally applicable, researchers have been developing more advanced feature selection methods based on a simple but potent idea: using multiple FS algorithms in a framework.

Hybrid Methods:

Hybrid methods implement different types of FS algorithms within one, multi-step process, taking advantage of their different characteristics (Fig.4).

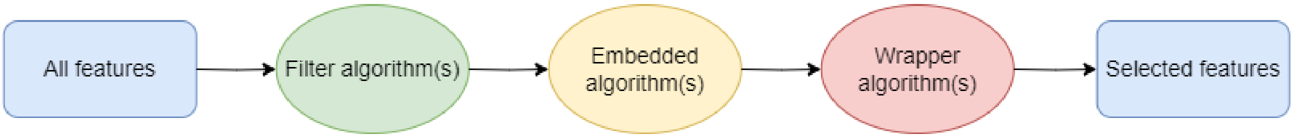

Figure 4: The process of feature selection with hybrid methods.

As Figure 4 shows, the most intuitive way to construct a hybrid method is to start with a fast filter technique and then give its (lower dimensional) output to a wrapper or embedded method, reducing their higher computational cost while retaining their higher accuracy – the best of both worlds [14].

Hybrid methods have become quite popular in bioinformatics, with some studies calling it the 'best practice' for feature selection [14, 15]. Jafari et al. [37] combined two univariate filters (Pearson correlation and information gain) and a multivariate filter (ReliefF) with a Genetic Algorithm wrapper to infer gene networks. Wang & Cai [38] analysed five different types of cancer with a two-step FS framework followed by an SVM classifier, manually confirming that the hybrid process selected near-optimal feature subsets.

Ensemble Methods:

Ensemble methods also utilize multiple FS algorithms, but unlike hybrid techniques, they do not implement them step-by-step, but rather, in parallel. In an ensemble process, multiple FS algorithms are run on the dataset separately, each of which returns a subset of features (Fig.5).

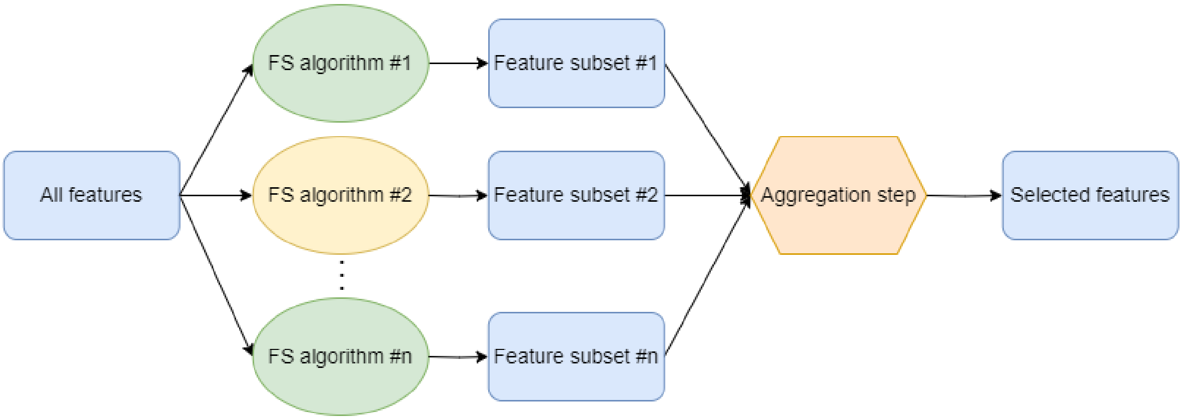

Figure 5: The process of feature selection with ensemble methods.

Then, these subsets are aggregated in some manner to find the final, ensemble feature set. This aggregation can be a simple intersection or union of the individual subsets or some kind of weighting of each feature based on its position in each individual subset [14].

Ensemble processes also increase the stability of feature selection [39], which is essential to generalise high performance across multiple datasets [36].

Verma et al. [18] used a variety of filter and embedded methods – wrappers are rarely present in ensembles – to show that different FS algorithms selected different features from genetic data, and thus an ensemble method is needed for the best performance. Farid et al. [40] proposed an ensemble feature selection and clustering method specifically for high-dimensional genomic data, showing that it worked better on a Brugada syndrome dataset than non-ensemble alternatives. Sarkar et al. [17] combined no less than eight different FS methods into an ensemble process and devised an innovative aggregation step that delivered high classification accuracy on breast cancer datasets.

ML Classifiers

As with our FS algorithms, we selected classifiers that are used with genomic data in the literature [9, 55], and which can perform well without requiring high computational power. The four algorithms we chose are: Random Forest (RF), Support Vector Machine (SVM), Naïve Bayes (NB), and Gradient Boosting (GB). They are all well-known and commonly used algorithms for a variety of machine learning tasks, thus we only give short descriptions of them in the following sections. All are available in and implemented from scikit-learn [44]. When possible, we tune the hyperparameters of the algorithm first to ensure a fair comparison. This tuning is done using grid search, which tests every possible combination of the given hyperparameter values, with 5-fold stratified cross-validation, similarly to Pasolli et al. [10] The process of the latter is the same as that of the 10-fold version detailed above, except here, the splits are made in a way that preserves the original distribution of classes in each split. The test metrics are averaged to give the final score for that particular combination of hyperparameters, and the combination that gives the highest accuracy is ultimately selected. The tuning is done separately for each tested dataset (using their full, unreduced versions) to ensure optimal performance.

Finally, we test each of the optimised classifiers individually on our datasets to find the best performing one to use in our ensemble framework.

Random Forest (RF)

We have already given a technical explanation of Random Forest in the feature selection methods section; here, we add that it is considered a state-of-the-art classification algorithm for metagenomic data [55, 56], which has been shown to outperform other algorithms like SVM, LASSO [57], and ENet [10]. It is well-suited for high dimensional datasets thanks to its in-built feature selection, and its tree-based structure makes its results interpretable.

The hyperparameters tuned for RF are the number of estimators (100, 200, 300, 400, 500) and the splitting criterion (entropy or Gini), both of which are identified as critical parameters for performance in LaPierre et al. [19] The rest are left at scikit-learn's default settings. The RF we use as an FS algorithm and the RF we use as the classifier always share the same hyperparameters for consistency.

Support Vector Machine (SVM)

The Support Vector Machine (SVM) is an algorithm that tries to separate data into classes by finding the hyperplane (also called the decision boundary) that has the largest margin between the support vectors: the different-class samples that are closest to this boundary on either side of it [9]. As only the support vectors are relevant for learning, SVM works well with high dimensional, low sample datasets like our metagenomic data, and its different kernels allow it to be effective on a wide variety of data. It is, accordingly, a popular choice for metagenomic classification [58, 59].

The hyperparameters tuned for SVM are the C regularisation parameter and the kernel coefficient gamma for the radial basis function (RBF) kernel we use. The tested values are taken from the MetAML study: $\{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$ for C and $\{2^{-15}, 2^{-13}, \ldots, 2^{3}\}$for gamma [10]. The rest are left at default settings.

Naïve Bayes (NB)

Naïve Bayes (NB) is a simple probabilistic classifier that applies Bayes' theorem with the assumption that the features are statistically independent from each other. This assumption is what makes it 'naïve.' Despite its simplicity, however, NB is used with genetic data, most notably in the field of taxonomic classification where it is the state-of-the-art algorithm [60].

Scikit-learn offers multiple Naïve Bayes implementations. Given that we are doing binary classification on continuous values, we chose Gaussian NB, which has no important hyperparameters to tune.

Gradient Boosting (GB)

Gradient Boosting (GB) is an ensemble machine learning technique that uses multiple weak learners, usually simple, fixed-size decision trees, and creates a prediction model from them by averaging their predictions [9]. As a boosting algorithm, it weighs the samples misclassified by previous learners higher than correctly predicted ones, which makes the next learners focus more on the misclassified samples. GB and its more advanced variants are used in metagenomic classification to great effect [61, 62]. It is also similar to RF, with both being decision tree-based ensemble classifiers – all in all, it is a worthwhile algorithm to include in our framework.

GB has the highest number of hyperparameters and requires the most tuning. Following the study of Bentéjac et al. [63], which extensively fine-tuned RF and boosting algorithms, our tuned hyperparameters are learning rate (0.025, 0.05, 0.1, 0.2, 0.3), maximum depth of each individual estimator (2, 3, 5, unlimited), and the minimum number of samples required to split an internal node (2, 5, 10). The rest are left at default settings.

EXPERIMENTS AND ANALYSIS

Dataset

In their landmark MetAML paper, Pasolli et al. [10] studied and made available six metagenomic datasets corresponding to five different conditions, with all samples taken from the human gut microbiome. The samples were gathered using shotgun sequencing, which provides a higher resolution to the level of microbial features and a higher consistency across studies compared to the older, more common 16S rRNA sequencing. We use these six datasets in our study. The datasets are organised into frequency tables: the rows are the samples, the columns are the (taxonomically identified) microbial features, and the values are the abundance of each feature in a given sample. We only use the species and strain-level (the latter is only available in certain cases) features. The samples are each labelled according to whether they belong to the case or control group. Some of these datasets have more than those two classes; we transform those into binary classification problems in the same way Pasolli et al. [10] did for better comparison with their study. The breakdown of these six datasets by condition, sample distribution, and feature count can be found in Table 2.

Table 2: Breakdown of the datasets used in the study

Dataset name	Disease	Case samples	Control samples	Total samples	Total features	Source study
Cirrhosis	Liver cirrhosis	118	114	232	562	[41]
CRC	Colorectal cancer	48	73	121	527	[2]
IBD	Inflammatory bowel diseases	173	319	492*	575	[42]
Obesity	Obesity	164	89	253	486	[6]
T2D	Type-2 diabetes	170	174	344	594	[43]
WT2D	Type-2 diabetes	53	43	96	431	[4]

Our datasets are varied in sample size, sample balance, and ease of classification – these are all important to test the robustness and generalisability of our algorithms. They cover a wide range of conditions from cancer to obesity, with two separate datasets available for type-2 diabetes (T2D sampled Chinese subjects, while WT2D sampled European women), which allows us to run a cross-study analysis as well.

In total, there are 1,538 samples available for analysis, providing a comprehensive base for our feature selection and classification experiments.

Applied FS Methods

We identified six FS algorithms used for disease detection from genomic data in the literature, which we found to be well-suited for usage with metagenomic data as well, given the similar challenges [24]. We implement five of those six algorithms here: Chi-squared (Chi2), Mutual Information (MI), Minimal-redundancy-maximal-relevance (mRMR), the ReliefF-based MultiSURF (MSURF), and Random Forest (RF). In addition, we include the Boruta wrapper algorithm in place of the previous study's FCBF, which we found to perform below par. Our reasoning for selecting the first five remains the same as in that study [24]: four are filter methods, which are frequently used in ensemble feature selection because of their speed and simplicity, while Random Forest is an embedded method, meaning it conducts feature selection

* Pasolli et al. [10] only considered 110 samples for IBD in their study; however, their published material included a second IBD dataset with 382 samples. We combined them to get our final sample count of 492. The other five datasets have the same sample size as in the original study.

and model training in the same step. Wrappers are not often included in ensemble feature selection because of their high computational cost which makes them infeasible on very high-dimensional data.

Chi-squared Test

The Chi-squared (Chi^2) test is a univariate, statistical based method applied to test the independence of two variables [14]. The test is considered valid if the test statistic follows a chi-squared distribution under the null hypothesis. There are multiple chi-squared tests, with the most common one being Pearson's chi-squared test, which is used to determine whether there is a statistically significant difference between the expected frequencies and the observed frequencies. We implemented the Chi-squared algorithm using the popular machine learning library scikit-learn [44].

Mutual Information

Mutual Information (MI) is another univariate statistical method, with a strong basis in information theory. It calculates the amount of information one variable gives about the other, thus measuring the dependency between them. It is considered more comprehensive than other measures of independence, as it is zero if and only if two features are fully independent [45], otherwise, it will be a non-negative value. The higher it is, the stronger the dependency, which makes it a fitting measure to use for feature selection. We used the scikit-learn [44] implementation of Mutual Information, which utilises entropy estimation from k-nearest neighbour distances as described in Kraskov et al. [45].

mRMR Algorithm

The mRMR (Minimal-redundancy-maximum-relevance) algorithm is a popular multivariate filter that has been effectively used for feature selection in several genomic big data studies [9, 15]. It is designed to solve the minimal-optimal problem: finding the smallest feature set with the highest predictive power. It is based on the previously described Mutual Information (MI) measure and built upon three concepts: Max-Dependency, Max-Relevance, and Min-Redundancy [46].

Max-Dependency is a scheme that aims to find the feature set that has the largest joint dependency on the target class. This is often difficult to accurately calculate, being downright infeasible on high-dimensional data [46], thus another criterion called Max-Relevance is used to approximate it. In Max-Relevance, features are considered individually, and the ones with the highest Mutual Information are selected. But choosing features based on relevance alone usually results in features with high redundancy [46], meaning that they depend on each other as well, not just the target variable. The Min-Redundancy condition is thus adopted to remove these redundant features and only keep the ones that are mutually exclusive [46]. The mRMR algorithm combines Max-Relevance and Min-Redundancy (hence its name) into one incremental technique to find the minimal-optimal feature set.

We use a popular, open-source Python implementation of mRMR, available at https://github.com/smazzanti/mrmr under the MIT license.

Relief and MultiSURF Algorithms

MultiSURF is a multivariate filter and one of the newer members of the Relief algorithm family. The original Relief algorithm has inspired several improvements and variants, most notably ReliefF, which has replaced it as the baseline Relief-based algorithm (RBA). ReliefF made multiple improvements (most notably implementing the user-specified k parameter for how many nearest neighbours the algorithm should find) and is able to deal with multi-class problems and missing data [48]. Another important variant is SURF, which used a distance threshold T instead of the previous k to define which instances are considered neighbours. SURF* iterated on SURF by dividing samples into 'near' and 'far' from the training sample and adjusting feature weighting accordingly [48]. MultiSURF* expanded on this by introducing a 'middle' distance zone; the samples within that zone were not included in the scoring. Finally, MultiSURF was introduced by Urbanowitz et al. [28] as an iteration on MultiSURF*: it preserved most of that algorithm but removed the 'far' scoring.

Urbanowitz et al. [28] also provided open-source implementations of all major RBAs, including MultiSURF, in a scikit-learn-compatible package called ReBATE, available https://github.com/EpistasisLab/scikit-rebate under the MIT license. This implementation of MultiSURF is the one we use.

Random Forest

Random Forest (RF) is an ensemble learning method and one of the most popular machine learning algorithms, widely used in numerous fields including metagenomic classification [9]. As it ranks every feature during learning, and these rankings can be easily retrieved, it is also considered an embedded feature selection method [14].

As an ensemble method, Random Forest uses multiple decision trees to learn the data, combining their predictions to produce one final result. The key is in how the decision trees are initialised and trained. RF utilises bagging: it selects a random sample of the data with replacement (meaning each sample can be selected more than once) to train each tree independently, meaning that each learner learns different data, thus reducing overfitting [49]. It also introduces a second layer of randomness with random feature selection: at each split, a decision tree only considers a subset of features instead of every feature, which reduces the correlation between each learner [49]. Once every tree has finished learning, their results are aggregated to produce the output of the Random Forest; in classification, this is a simple majority vote. During training, each decision tree calculates the importance of the features it encounters (there are multiple criteria that can be used; for example, scikit-learn offers Gini impurity and information gain [44]), and in the end, these importances are averaged across every tree and normalised to produce the final score for that feature. These scores can be extracted from the model and used to rank the features, similarly to a filter.

Boruta

Boruta is a wrapper algorithm designed to solve the all-relevant feature selection problem: to find every feature that is relevant for classification, rather than just the feature set that gives the highest classification performance (which is the minimal-optimal problem mentioned in our description of mRMR) [50]. The main steps of the algorithm are as follows:

1. For each feature in a dataset, the algorithm creates a so-called shadow feature, with its values taken from the original feature but randomised so it is not correlated to the target.
2. Next, Random Forest is run on the combined dataset of features and shadow features. The shadow feature with the highest Z score is taken as the threshold.
3. Real features with significantly higher importance than the threshold are confirmed as important. Features with significantly lower importance are discarded. The rest are left undecided for now.
4. The shadow features are removed, and new ones are generated. The steps repeat until either all features are assigned or the algorithm hits the iteration limit set by the user.

As a wrapper algorithm, Boruta is much more computationally intensive than our filters – indeed, the creators of the algorithm noted that its goal is not to reduce computational time [50]. But studies have demonstrated its efficacy in metagenomic feature selection [51, 52], and its all-relevant nature fits especially well into our ensemble feature selection framework, as finding every important feature instead of just the minimal-optimal set is better for seeing how much its findings align with that of other algorithms, which it will be aggregated with.

We use the best-known Python implementation of Boruta, known as BorutaPy, available at https://github.com/scikit-learn-contrib/boruta_py under the BSD-3-Clause license.

Individual FS Selection

To test each of our feature selection (and ML classification) algorithms, we first run experiments with each FS method individually, on every dataset, with varying subset sizes. Our filter and embedded methods rank every feature, and we select the highest ranked ones to form that method's feature subset. The subset sizes range from 10 to 200 – the upper bound is set as such because our datasets generally have 400-500 features in total, thus selecting more than 200 features would not effectively reduce the dimensionality of the dataset, which is one of the main goals of feature selection. Note that Boruta, as a wrapper, always selects what it deems is the optimal feature subset, thus we cannot test different subset sizes with it.

In each experiment, the acquired feature subset is used to train and test one of our ML classifiers, and we store the resulting AUC and accuracy scores (as well as the feature count and the names of the dataset, the classifier, and the FS algorithm) for analysis. The subset with the highest AUC is furthermore chosen as that FS algorithm's contribution to the ensemble, which will be aggregated with the subsets of the other algorithms chosen the same way.

It is also at this step that we find the best classifier. We run the individual FS experiments with all subset sizes on three datasets: Cirrhosis, T2D, and Obesity. We chose these three largely because they differ in 'classification difficulty,' as shown by Pasolli et al. [10] and confirmed by our own preliminary experiments. Cirrhosis is one of the easiest to classify, with the highest AUC scores regardless of algorithm, while Obesity is the hardest, with the lowest scores. T2D lies in between, while also being our largest balanced dataset. Together, these three provide a strong basis to select the best-performing classifier which we will use in our ensemble experiments.

Feature Aggregation

After each FS algorithm has produced a feature subset, these must be aggregated into one final, ensemble feature set which the final ML model can be trained on. This aggregation can be done in several different ways. In this study, we consider three: union, consensus (or simple voting), and weighted voting.

In the union method, we select every feature that at least one FS algorithm has selected. Verma et al. [18] found that taking the union of features selected by multiple methods preserved important features better than single algorithms on genetic data. However, depending on the heterogeneity of the selections, the size of the union set might not be much smaller than the total number of features and might not alleviate the curse of dimensionality.

The consensus or simple voting method labels each feature according to how many algorithms selected it, from one to six – the same technique Sarkar et al. [17] used in their ensemble feature selection study. The more algorithms selected a feature, the more informative it is deemed to be. It is important to mention that this method treats every FS algorithm equally, with an equal vote, which might be suboptimal if there are large differences in classification performance between them.

To solve this issue, we have also devised a weighted voting aggregation scheme that takes into account the performances of the participating FS algorithms. As before, each FS method selects the number of features which deliver the subset with the highest classification AUC score. This value is recorded and serves as that algorithm's vote weight, making it so that better-performing algorithms have a stronger say in which features get selected for the ensemble. Specifically, each feature from the union set (the others are discarded as no algorithm selected them; therefore, they are unlikely to be informative) is given a score: the sum of the squared vote weights of the algorithms that selected that feature. We have chosen to square the values to emphasise the difference between stronger and weaker algorithms. After scoring each feature, we create three ensemble sets from the top 30%, 20%, and 10% of features, respectively.

The Framework

A summary visualisation of our framework can be seen in Figure 6. The exact algorithms, steps, and configurations that constitute it are detailed in the sections below.

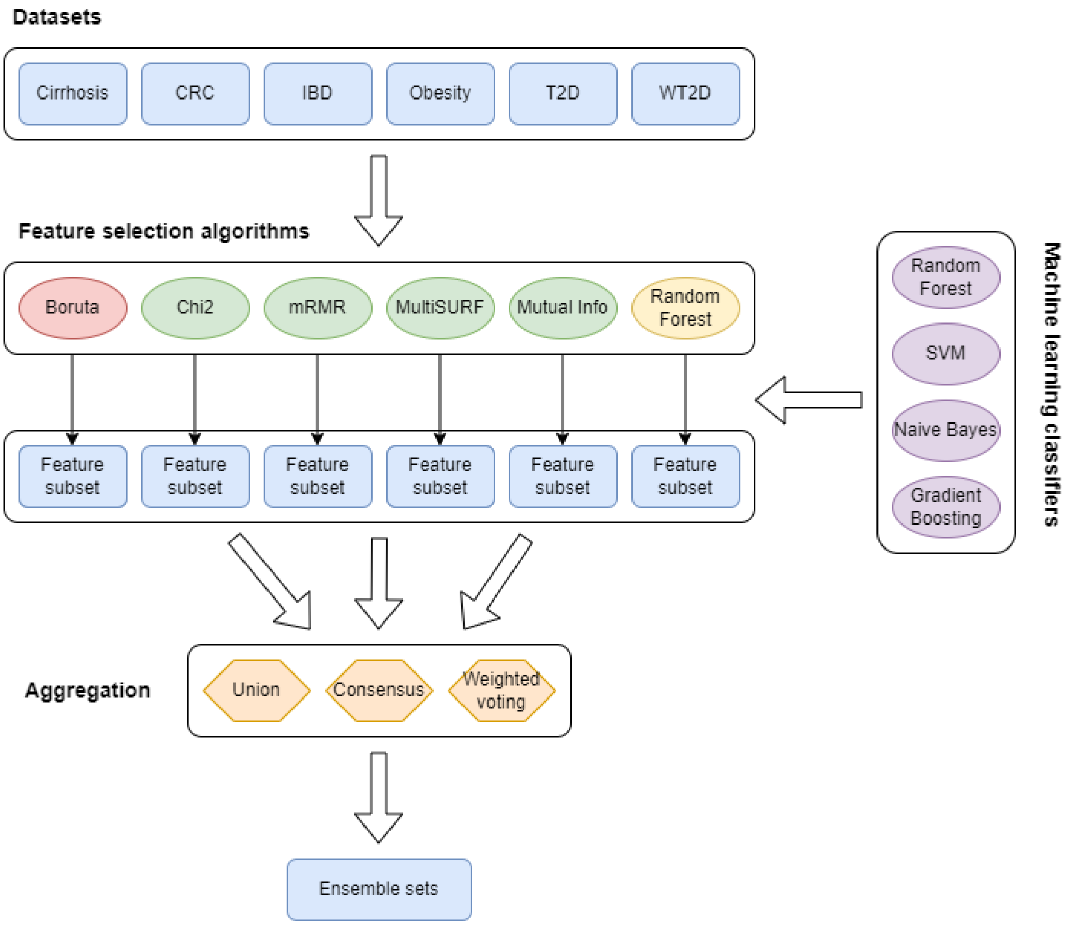

Figure 6: The ensemble framework.

RESULTS

Individual Feature Selection

Selecting the Best ML Classifier

First, we tested each, selected FS algorithm and subset size combination with the Cirrhosis, T2D, and Obesity datasets to find which classifier performed best. The baseline (without feature selection) and the best FS results (best meaning highest AUC score in this case) for each can be found in Table 3.

Table 3: AUC and accuracy scores (with standard deviations in parentheses) of the four tested ML classifiers with and without feature selection. The highest performing classifier for each dataset is bolded.

		Baseline score		Best FS score		
Dataset	**Classifier**	**AUC**	**Accuracy**	**AUC**	**Accuracy**	**Best FS algo.**
Cirrhosis	**RF**	**0.951 (0.042)**	**0.875 (0.067)**	**0.964 (0.034)**	**0.862 (0.056)**	**RF**
	SVM	0.826 (0.088)	0.742 (0.088)	0.93 (0.039)	0.608 (0.062)	MI

	NB	0.770 (0.090)	0.759 (0.098)	0.911 (0.039)	0.819 (0.06)	mRMR
	GB	0.896 (0.055)	0.814 (0.034)	0.93 (0.053)	0.862 (0.06)	MI
T2D	RF	0.726 (0.070)	0.678 (0.058)	0.795 (0.076)	0.721 (0.072)	Boruta
	SVM	0.727 (0.112)	0.666 (0.090)	0.755 (0.103)	0.683 (0.077)	mRMR
	NB	**0.623 (0.089)**	**0.622 (0.088)**	**0.838 (0.063)**	**0.75 (0.068)**	**mRMR**
	GB	0.682 (0.064)	0.655 (0.048)	0.748 (0.077)	0.672 (0.069)	Boruta
Obesity	**RF**	**0.647 (0.070)**	**0.656 (0.034)**	**0.759 (0.094)**	**0.714 (0.093)**	**mRMR**
	SVM	0.543 (0.129)	0.648 (0.012)	0.711 (0.096)	0.661 (0.046)	mRMR
	NB	0.529 (0.099)	0.513 (0.115)	0.721 (0.08)	0.719 (0.034)	mRMR
	GB	0.576 (0.105)	0.652 (0.049)	0.708 (0.061)	0.68 (0.067)	Boruta

Random Forest performed best on the Cirrhosis and Obesity datasets, with the highest feature selected AUC scores. It had the highest or almost-highest baseline scores in all three, which can be attributed to the fact that it already conducts some feature selection implicitly during the training process, making it a better fit for high-dimensional data than algorithms without this step. This performance is in-line with the literature [55,56,57] and further confirms that RF is a premier choice for metagenomic classification.

Naïve Bayes outperformed RF in both metrics on the T2D dataset and in accuracy on Obesity. This is an impressive result, given that NB is a simple algorithm [58] and was the only one not fine-tuned before classification. NB classifiers have long dominated the field of taxonomic classification of gene sequences [60]; these results suggest that they are effective in other domains where taxonomic data is involved as well.

The other two tested classifiers, SVM and Gradient Boosting, performed considerably worse. This, however, should not be taken as a suggestion that these algorithms are not effective for metagenomic classification. As we set out in the goals of this study, we are working with the scikit-learn implementations of all these classifiers, which are not designed for bleeding-edge performance but for wide accessibility and low computational resource usage.

The results here also give an initial idea of which FS algorithms the classifiers work best with. There is some variance in the Cirrhosis dataset, though because of the overall high AUC values, there are often only minuscule (within the standard deviation) differences between the scores of the algorithms. In the other two, mRMR and Boruta (and especially the former) dominate. According to our result and the literature review, we chose Random Forest because of its high performance on all tested datasets, its in-built feature selection, and the interesting insights that can be gleaned from using it both as the classifier and as an FS algorithm. All the scores in the rest of our results were acquired using RF as the classifier.

Comparing Feature Selection Algorithms

We conducted hundreds of experiments, running all six datasets through each of our FS algorithms 20 times, each time with a different subset size ranging from 10 to 200. The goal was to get a clear picture of the performance of our chosen algorithms, which is important for the aggregation step that follows. More generally, we also wished to contribute to the literature by providing a wide set of results not previously available. Below are the tables with the scores per subset size for each FS algorithm for the CRC and Obesity datasets. The underlying data for these and the other four datasets are available as part of the Supplementary Material attached to this study.

Table 4: Baseline scores of the tuned RF classifier being trained and tested on the full, unreduced datasets.

Dataset	AUC	Accuracy
Cirrhosis	0.951 (0.042)	0.875 (0.067)
CRC	0.871 (0.129)	0.801 (0.120)
IBD	0.946 (0.061)	0.864 (0.074)
Obesity	0.647 (0.070)	0.656 (0.034)
T2D	0.726 (0.070)	0.678 (0.058)
WT2D	0.719 (0.148)	0.688 (0.111)

Table 5: AUC scores (with standard deviations in parentheses) of the six FS algorithms with all tested subset sizes on the CRC dataset. The baseline score (on the full dataset with all 527 features) is included for comparison. The highest score for each algorithm is bolded; in case of a tie, only the lowest feature count is bolded. Boruta selected 14 features.

Feature count	Chi^2	mRMR	MultiSURF	Mutual Info	RF	Boruta (14)
Baseline (527)	0.871 (0.129)					
10	0.854 (0.112)	0.835 (0.095)	0.746 (0.202)	0.862 (0.083)	0.901 (0.108)	**0.923 (0.076)**
20	0.881 (0.087)	0.874 (0.138)	0.767 (0.116)	0.864 (0.119)	0.911 (0.067)	
30	0.888 (0.108)	0.899 (0.118)	0.87 (0.092)	0.896 (0.101)	0.913 (0.085)	
40	**0.908 (0.092)**	0.906 (0.116)	0.879 (0.09)	0.886 (0.138)	**0.925 (0.075)**	
50	0.897 (0.093)	0.895 (0.121)	0.875 (0.093)	0.887 (0.129)	0.909 (0.085)	
60	0.898 (0.103)	0.897 (0.122)	0.868 (0.106)	**0.911 (0.116)**	0.894 (0.102)	
70	0.894 (0.094)	0.906 (0.118)	**0.916 (0.085)**	0.904 (0.116)	0.894 (0.13)	
80	0.903 (0.094)	0.886 (0.115)	0.905 (0.082)	0.868 (0.155)	0.912 (0.091)	
90	0.908 (0.087)	0.889 (0.116)	0.902 (0.097)	0.889 (0.141)	0.896 (0.112)	
100	0.9 (0.096)	0.906 (0.108)	0.895 (0.121)	0.898 (0.146)	0.889 (0.126)	
110	0.896 (0.11)	0.912 (0.111)	0.915 (0.086)	0.901 (0.156)	0.9 (0.105)	
120	0.903 (0.101)	0.915 (0.097)	0.898 (0.096)	0.898 (0.103)	0.9 (0.097)	
130	0.908 (0.078)	0.924 (0.093)	0.909 (0.103)	0.894 (0.137)	0.895 (0.113)	
140	0.883 (0.117)	0.926 (0.094)	0.907 (0.107)	0.888 (0.135)	0.886 (0.116)	
150	0.892 (0.101)	**0.929 (0.097)**	0.879 (0.128)	0.895 (0.146)	0.889 (0.121)	
160	0.882 (0.11)	0.928 (0.079)	0.892 (0.105)	0.886 (0.153)	0.883 (0.114)	
170	0.888 (0.11)	0.918 (0.105)	0.901 (0.105)	0.883 (0.143)	0.877 (0.129)	
180	0.895 (0.121)	0.918 (0.106)	0.888 (0.123)	0.886 (0.132)	0.892 (0.115)	
190	0.893 (0.111)	0.918 (0.105)	0.892 (0.107)	0.898 (0.154)	0.889 (0.103)	
200	0.886 (0.117)	0.912 (0.11)	0.898 (0.106)	0.899 (0.13)	0.884 (0.128)	

Table 6: AUC scores (with standard deviations in parentheses) of the six FS algorithms with all tested subset sizes on the Obesity dataset. Boruta selected 7 features.

Feature count	Chi^2	mRMR	MultiSURF	Mutual Info	RF	Boruta (7)
Baseline (486)	0.647 (0.070)					
10	0.614 (0.098)	0.67 (0.096)	0.662 (0.12)	0.559 (0.119)	0.678 (0.064)	**0.683 (0.09)**
20	0.641 (0.081)	0.7 (0.092)	0.678 (0.11)	0.589 (0.142)	0.703 (0.081)	
30	0.646 (0.096)	0.733 (0.067)	0.678 (0.128)	0.64 (0.078)	0.712 (0.074)	
40	0.657 (0.096)	0.752 (0.08)	0.673 (0.116)	**0.682 (0.075)**	**0.715 (0.103)**	
50	0.646 (0.123)	0.732 (0.094)	0.668 (0.126)	0.625 (0.101)	0.711 (0.101)	
60	0.668 (0.105)	0.738 (0.097)	0.669 (0.115)	0.647 (0.138)	0.689 (0.086)	
70	0.658 (0.105)	0.758 (0.087)	0.669 (0.131)	0.676 (0.089)	0.692 (0.09)	
80	0.663 (0.094)	**0.759 (0.094)**	0.666 (0.144)	0.634 (0.09)	0.666 (0.103)	
90	0.668 (0.092)	0.736 (0.084)	0.676 (0.117)	0.653 (0.117)	0.7 (0.112)	
100	**0.671 (0.103)**	0.723 (0.091)	0.673 (0.12)	0.657 (0.105)	0.687 (0.098)	
110	0.645 (0.097)	0.73 (0.107)	0.67 (0.111)	0.658 (0.12)	0.675 (0.085)	
120	0.656 (0.114)	0.736 (0.093)	0.664 (0.121)	0.677 (0.125)	0.669 (0.133)	
130	0.631 (0.13)	0.741 (0.069)	0.676 (0.096)	0.653 (0.107)	0.677 (0.113)	
140	0.644 (0.117)	0.723 (0.082)	**0.703 (0.099)**	0.655 (0.108)	0.656 (0.1)	
150	0.668 (0.089)	0.726 (0.074)	0.684 (0.093)	0.677 (0.101)	0.678 (0.09)	
160	0.646 (0.116)	0.715 (0.063)	0.676 (0.084)	0.643 (0.109)	0.656 (0.102)	
170	0.647 (0.109)	0.716 (0.075)	0.688 (0.09)	0.671 (0.143)	0.669 (0.102)	
180	0.639 (0.129)	0.71 (0.1)	0.692 (0.106)	0.648 (0.123)	0.654 (0.063)	
190	0.634 (0.115)	0.731 (0.078)	0.671 (0.109)	0.62 (0.114)	0.651 (0.076)	
200	0.655 (0.12)	0.723 (0.088)	0.701 (0.094)	0.661 (0.107)	0.648 (0.1)	

These results, as well as those we saw on the other datasets, unanimously speak to the effectiveness of feature selection: on every dataset, with every FS algorithm, there is an improvement in AUC (and accuracy) over the baseline without feature selection. This is well in-line with the literature; as we discussed in the Background section, feature selection is acknowledged as a necessary step in the classification of metagenomic data. It is important to once again note that with RF being our classifier, some feature selection is done even on the baseline – but additional, explicit feature selection improves scores even further.

The magnitude of this improvement depends on multiple factors, however. First is the dataset itself: generally, datasets that are harder to classify, with lower baseline scores (such as Obesity and WT2D) see greater increases in AUC after feature selection than datasets with high baselines. The subset size also plays a large role: the tables show that the optimal subset size differs per algorithm and per dataset alike, meaning there is no one-size-fits-all number. In fact, a poor choice of subset size can bring the performance of an algorithm down to or even below the baseline level. This is an advantage of wrapper methods like Boruta: while its performance is not greater than our best filters, it only requires a single run to achieve it, though of course at a higher computational cost. For the rest, testing multiple different subset sizes is essential to achieve the best performance. The range of sizes that should be tested depends on the number of features in the given dataset, though the results show that reducing the feature set too much (leaving less than 5% of the features) is detrimental, as is keeping it too large (leaving more than 30% of the features). This makes intuitive sense: if the dimensionality reduction is too drastic, some important features will inevitably get lost, but if it is not significant enough, then irrelevant features and noise will remain in the dataset and reduce classification performance.

Ensemble Feature Selection

As detailed in our Methodology section, we applied three different types of subset aggregation: union, consensus, and weighted voting. All three methods were applied on the feature subsets generated for every dataset, and the resulting ensemble sets were tested with the same tuned RF classifier that we used for the individual experiments to ensure comparability. First, we state the results of the dimensionality reduction these methods achieved, then their prediction performance.

Dimensionality Reduction Results

Table 7: The tested aggregation steps and the number of features in their ensemble sets for each dataset, with the percentage of the dimensionality reduction in parentheses.

		Cirrhosis	CRC	IBD	Obesity	T2D	WT2D
	Total features	562	527	575	486	594	431
Union	**Union**	299 (46.8%)	227 (56.9%)	349 (39.3%)	261 (46.3%)	310 (47.8%)	253 (41.3%)
Consensus	**At least in 2 subsets**	150 (73.3%)	79 (85%)	171 (70.3%)	98 (79.8%)	145 (75.6%)	112 (74%)
	At least in 3 subsets	84 (85.1%)	36 (93.2%)	93 (83.8%)	36 (92.6%)	60 (89.9%)	33 (92.3%)
	At least in 4 subsets	54 (90.4%)	18 (96.6%)	58 (89.9%)	8 (98.4%)	20 (96.6%)	13 (97%)
	At least in 5 subsets	32 (94.3%)	12 (97.7%)	41 (92.9%)	4 (99.2%)	9 (98.5%)	4 (99.1%)
	At least in 6 subsets	17 (97%)	2 (99.6%)	23 (96%)	-	2 (99.7%)	-
Weighted voting	**Top 30% of features**	96 (82.9%)	72 (86.3%)	107 (81.4%)	83 (82.9%)	114 (80.8%)	77 (82.1%)
	Top 20% of features	62 (89%)	48 (90.9%)	78 (86.4%)	59 (87.9%)	62 (89.6%)	54 (87.5%)
	Top 10% of features	31 (94.5%)	29 (94.5%)	37 (93.6%)	27 (94.4%)	34 (94.3%)	28 (93.5%)

The union sets naturally have the highest dimension, including between 45% to 60% of a dataset's features. However, more than half of these features were selected by only one algorithm; this holds true for all datasets, even the easier-to-classify ones like Cirrhosis and IBD. These results further reinforce the finding of previous studies that metagenomic data is unstable: different FS algorithms will choose different features as informative [36, 18]. With the exception of the IBD dataset (which has the most samples by a considerable margin), the union sets were larger than the number of samples in each dataset, meaning the reduction was not enough to overcome the curse of dimensionality.

The consensus sets highlight the level of heterogeneity between the individual selections, or in other words, how much the individual algorithms agreed on the important features. As expected, harder-to-classify datasets have higher heterogeneity, though the severity of it is still notable: There are zero features in Obesity and WT2D (and only two in T2D) that were selected by all six FS algorithms, while less than 4% of features in these datasets were selected by four. This shows a downside of the consensus method: without advance knowledge of (or preliminary experiments on) a dataset, it is difficult to gauge the level of filtering required for the best results. It is intuitive to want to keep only the features selected by almost every or every algorithm, which will work on some datasets but will leave too few features, if any, on others. The ensemble sets from the weighted voting method do not have this issue as their sizes are based on the union set, which can be expected to always be large given the heterogeneity of individual selections.

Classification Performance Results

Table 8: The best performing (highest AUC) ensemble methods for each dataset, compared to the best performing individual algorithms.

Dataset	Best ensemble method	Feature count	AUC	Accuracy	Best indiv. AUC	Best indiv. algorithm
Cirrhosis	At least in 3 subsets	84	0.962 (0.034)	0.875 (0.064)	0.964 (0.034)	Random Forest
CRC	At least in 5 subsets	12	0.929 (0.072)	0.843 (0.126)	0.929 (0.097)	mRMR
IBD	At least in 4 subsets	58	0.954 (0.063)	0.866 (0.079)	0.962 (0.05)	Chi2
Obesity	Top 30% of features	83	0.727 (0.096)	0.692 (0.045)	0.759 (0.094)	mRMR
T2D	Top 30% of features	114	0.768 (0.064)	0.686 (0.061)	0.795 (0.076)	Boruta
WT2D	At least in 4 subsets	13	0.878 (0.088)	0.804 (0.102)	0.851 (0.116)	Mutual Info

At the end of this multi-step ensemble framework, we arrived at the best ensemble sets for each dataset, listed in Table 8. The method that was deemed the best differs between the datasets, but crucially, they perform comparably well to the best single FS algorithms. The largest differences are in the Obesity (an AUC reduction of 0.032) and T2D (reduction of 0.027) datasets, but they are both well within the standard deviation of the AUC scores. In the other datasets, there is virtually no difference between the best ensemble and the best single algorithm, while the ensemble actually beat out the individual algorithm in the WT2D dataset.

The performance of these ensemble sets is better than what Pasolli et al. [10] were able to achieve in their original study. They considered two different kinds of data, species abundance (which is what we also use) and strain-specific marker presence, and they found that using the latter improved prediction performance in five of the six datasets. Our ensemble framework delivered higher AUC values than their best marker presence scores in five datasets, with Cirrhosis being essentially equal, despite using the less-precise species abundance data. Other studies have worked with the MetAML datasets since, with LaPierre et al. [19] aggregating the results of three deep learning approaches: PopPhy [23], Met2Img [64], and RegMIL [65]. We also add DeepMicro, another state-of-the-art [22] neural network framework using the MetAML data [21]. The comparison of their results with ours on some of the datasets (as not every framework processed every dataset) can be found in Table 9.

Table 9: Comparison of metagenomic frameworks (including ours) on four of the six datasets. The values for PopPhy, Met2Img, and RegMIL are taken from the review of LaPierre et al. [19] For MetAML, the scores using species abundance are listed. The best scores for each dataset and metric are bolded.

	Cirrhosis		T2D		Obesity		IBD	
Framework	**Accuracy**	**AUC**	**Accuracy**	**AUC**	**Accuracy**	**AUC**	**Accuracy**	**AUC**
MetAML-SVM	0.834 (0.052)	0.922 (0.041)	0.613 (0.057)	0.663 (0.066)	0.636 (0.042)	0.648 (0.071)	0.809 (0.066)	0.862 (0.083)
MetAML-RF	0.877 (0.043)	0.945 (0.036)	0.664 (0.052)	0.744 (0.056)	0.644 (0.028)	0.655 (0.079)	0.809 (0.050)	0.890 (0.078)
PopPhy-RF	NA	0.932	NA	0.727	NA	0.642	NA	NA
PopPhy-CNN	NA	0.94	NA	0.753	NA	0.676	NA	NA
Met2Img-RF	0.877 (0.060)	NA	0.672 (0.080)	NA	0.645 (0.042)	NA	0.808 (0.068)	NA
Met2Img-CNN	0.905 (0.071)	NA	0.651 (0.094)	NA	0.680 (0.066)	NA	**0.868 (0.081)**	NA
RegMIL baseline	0.923 (0.041)	0.922 (0.040)	NA	NA	NA	NA	0.8387 (0.028)	0.8242 (0.0374)
RegMIL-RF	**0.928 (0.036)**	0.927 (0.035)	NA	NA	NA	NA	0.847 (0.035)	0.844 (0.026)
DeepMicro	NA	0.940	NA	0.763	NA	0.659	NA	**0.955**
Our framework	0.875 (0.064)	**0.962 (0.034)**	**0.686 (0.061)**	**0.768 (0.064)**	**0.692 (0.045)**	**0.727 (0.096)**	0.866 (0.079)	0.954 (0.063)

It is important to note that drawing conclusions based on these scores alone is difficult because of the differing hyperparameters, feature counts, and cross-validation strategies used in the studies [19]. Nevertheless, we can say that our results validate the utility of ensemble methods and the strength of our framework. When given a new sample or dataset, it is not possible to know in advance which FS algorithm will work best on it – and it is well-established that no one algorithm will be ideal in every case. Using an ensemble method with multiple similarly strong FS algorithms eliminates this problem, creating a generalisable framework applicable to any (metagenomic) data, without much, if any, loss of classification performance. It alleviates the instability inherent in this type of data while also reducing dimensionality (in the case of the consensus and weighted voting ensembles, at least) below the number of samples, making metagenomic datasets better fit for machine learning.

While ensembling solved the issue of having to choose between algorithms, there are still multiple aggregation methods to choose from. There is no clear favourite, but based on our overall findings, we recommend taking the top 30% of features by weighted score if one method must be chosen. This method performed best on the two hardest-to-classify datasets, while on every other dataset, it performed on the level of the best methods, which is shown in Table 10. Furthermore, we believe that the top 30% method strikes a good balance in terms of dimensionality reduction: its ~80% reduction rate is enough to weed out noise and most irrelevant features, without reducing the feature set to the point where informative features are likely lost. By definition, it is also much less dependent on the overall classification difficulty of the given dataset than the consensus method.

Table 10: The performance of the top 30% features method, compared to the best ensemble and individual algorithms for each dataset.

	Top 30% of features method			
Dataset	**Feature count**	**AUC**	**Difference to best ensemble**	**Difference to best individual**
Cirrhosis	96	0.957 (0.042)	0.005	0.007
CRC	72	0.908 (0.107)	0.021	0.021
IBD	107	0.953 (0.059)	0.001	0.009
Obesity	83	0.727 (0.096)	-	0.032
T2D	114	0.768 (0.064)	-	0.027
WT2D	77	0.864 (0.077)	0.014	-0.013

CONCLUSION

We built an ensemble feature selection and machine learning classification framework for metagenomic data. To our knowledge, there hasn't been a study before that implemented ensemble feature selection specifically on metagenomic data and used all six of the benchmark MetAML datasets to test it. Existing metagenomic frameworks for disease prediction are mostly deep learning-based, requiring powerful computational resources, while our solution solely utilises open-source algorithms that can be run on consumer hardware, even without any GPU acceleration. Despite its simpler structure, our framework performs comparably with the state-of-the-art methods, even outperforming them in certain cases, though the number of variables involved makes direct comparisons difficult.

Consistent with the literature, we found that feature selection universally improves classification performance, though this improvement varies per dataset and, on non-wrapper methods, depends on choosing the right subset size. When looking at their best scores, the six FS algorithms performed broadly similarly across the data, with the largest differences being on the hardest-to-classify datasets, where mRMR and Boruta edged out the others. Despite their similar performance, we found that each algorithm selected different feature subsets, with more than half of the union set's features only being selected by one algorithm. This highlights the complexity of metagenomic data and the need for an ensemble methodology to cover algorithmic blind spots and make feature selection more stable.

By testing four popular and commonly used ML classifiers, we confirmed that RF remains the premier algorithm to use with metagenomic data thanks to its high baseline performance, built-in feature selection, and interpretable nature. Among ensemble methods, the consensus and weighted voting methods both achieved sufficient dimensionality reduction, and their classification performance was up to par with the best individual FS algorithms. We highlighted the method of taking the top 30% features by weighted score as a generally high-performing, adaptive aggregation step that does especially well on hard-to-classify datasets. Using our ensemble methods, we were also able to identify several biomarkers that were important signifiers of more than one disease, contributing to the growing literature of microbial features identified through machine learning methods.

Our ensemble framework was competitive with other methods that use metagenomic data (and specifically, the MetAML datasets) for disease prediction, but it did not bring any breakthroughs in classification performance. While our focus was more on simplicity and accessibility than bleeding-edge performance, it is also possible that we are approaching the limits of what be achieved using solely metagenomic (and more specifically, species abundance) data. This was already raised as a possibility some years ago [19], and researchers have been exploring using other forms of metagenomic data or combining it with different types of information to break this plateau. One example are strain-specific markers, employed already in the original MetAML study [10], which might better describe complex diseases, at the cost of being drastically higher dimensional than species abundance. Other studies have looked at augmenting metagenomic data with different genetic data (such as metabolomic profiles) [22] or demographic attributes [74], both promising avenues. We plan on investigating these areas in future work, while also looking into improving the (for now, lacklustre) generalisation capabilities of the framework between different datasets for the same disease.

To close this study, we would like to emphasise the value of open-source data and algorithms, which we believe are essential for a thriving research community in metagenomics and beyond. To this end, we are glad to provide our ensemble framework and our wide range of experimental data to the public as well, and we hope it will aid and inspire further research in the field.

References

[1] J. C. Wooley, A. Godzik and I. Friedberg, "A Primer on Metagenomics," *PLoS Computational Biology,* vol. 6, p. e1000667, February 2010.

[2] G. Zeller, J. Tap, A. Y. Voigt, S. Sunagawa, J. R. Kultima, P. I. Costea, A. Amiot, J. Böhm, F. Brunetti, N. Habermann, R. Hercog, M. Koch, A. Luciani, D. R. Mende, M. A. Schneider, P. Schrotz-King, C. Tournigand, J. T. V. Nhieu, T. Yamada, J. Zimmermann, V. Benes, M. Kloor, C. M. Ulrich, M. K. Doeberitz, I. Sobhani and P. Bork, "Potential of fecal microbiota for early-stage detection of colorectal cancer," *Molecular Systems Biology,* vol. 10, p. 766, November 2014.

[3] Y. Zhang, J. Shen, X. Shi, Y. Du, Y. Niu, G. Jin, Z. Wang and J. Lyu, "Gut microbiome analysis as a predictive marker for the gastric cancer patients," *Applied Microbiology and Biotechnology,* vol. 105, p. 803–814, January 2021.

[4] F. H. Karlsson, V. Tremaroli, I. Nookaew, G. Bergström, C. J. Behre, B. Fagerberg, J. Nielsen and F. Bäckhed, "Gut metagenome in European women with normal, impaired and diabetic glucose control," *Nature,* vol. 498, p. 99–103, May 2013.

[5] T. R. Abrahamsson, H. E. Jakobsson, A. F. Andersson, B. Björkstén, L. Engstrand and M. C. Jenmalm, "Low gut microbiota diversity in early infancy precedes asthma at school age," *Clinical & Experimental Allergy,* vol. 44, p. 842–850, May 2014.

[6] E. Le Chatelier, T. Nielsen, J. Qin, E. Prifti, F. Hildebrand, G. Falony, M. Almeida, M. Arumugam, J.-M. Batto, S. Kennedy, P. Leonard, J. Li, K. Burgdorf, N. Grarup, T. Jørgensen, I. Brandslund, H. B. Nielsen, A. S. Juncker, M. Bertalan, F. Levenez, N. Pons, S. Rasmussen, S. Sunagawa, J. Tap, S. Tims, E. G. Zoetendal, S. Brunak, K. Clément, J. Doré, M. Kleerebezem, K. Kristiansen, P. Renault, T. Sicheritz-Ponten, W. M. de Vos, J.-D. Zucker, J. Raes, T. Hansen, E. Guedon, C. Delorme, S. Layec, G. Khaci, M. van de Guchte, G. Vandemeulebrouck, A. Jamet, R. Dervyn, N. Sanchez, E. Maguin, F. Haimet, Y. Winogradski, A. Cultrone, M. Leclerc, C. Juste, H. Blottière, E. Pelletier, D. LePaslier, F. Artiguenave, T. Bruls, J. Weissenbach, K. Turner, J. Parkhill, M. Antolin, C. Manichanh, F. Casellas, N. Boruel, E. Varela, A. Torrejon, F. Guarner, G. Denariaz, M. Derrien, J. E. T. van Hylckama Vlieg, P. Veiga, R. Oozeer, J. Knol, M. Rescigno, C. Brechot, C. M'Rini, A. Mérieux, T. Yamada, P. Bork, J. Wang, S. D. Ehrlich and O. Pedersen, "Richness of human gut microbiome correlates with metabolic markers," *Nature,* vol. 500, p. 541–546, August 2013.

[7] A. Oulas, C. Pavloudi, P. Polymenakou, G. A. Pavlopoulos, N. Papanikolaou, G. Kotoulas, C. Arvanitidis and loannis Iliopoulos, "Metagenomics: Tools and Insights for Analyzing Next-Generation Sequencing Data Derived from Biodiversity Studies," *Bioinformatics and Biology Insights,* vol. 9, p. BBI.S12462, January 2015.

[8] K. Katz, O. Shutov, R. Lapoint, M. Kimelman, J. Brister and C. O'Sullivan, "The Sequence Read Archive: a decade more of explosive growth," *Nucleic Acids Research,* vol. 50, p. D387–D390, November 2021.

[9] L. J. Marcos-Zambrano, K. Karaduzovic-Hadziabdic, T. L. Turukalo, P. Przymus, V. Trajkovik, O. Aasmets, M. Berland, A. Gruca, J. Hasic, K. Hron, T. Klammsteiner, M. Kolev, L. Lahti, M. B. Lopes, V. Moreno, I. Naskinova, E. Org, I. Paciência, G. Papoutsoglou, R. Shigdel, B. Stres, B. Vilne, M. Yousef, E. Zdravevski, I. Tsamardinos, E. C. de Santa Pau, M. J. Claesson, I. Moreno-Indias and J. Truu, "Applications of Machine Learning in Human Microbiome Studies: A Review on Feature Selection, Biomarker Identification, Disease Prediction and Treatment," *Frontiers in Microbiology,* vol. 12, February 2021.

[10] E. Pasolli, D. T. Truong, F. Malik, L. Waldron and N. Segata, "Machine Learning Meta-analysis of Large Metagenomic Datasets: Tools and Biological Insights," *PLOS Computational Biology,* vol. 12, p. e1004977, July 2016.

[11] M. Oudah and A. Henschel, "Taxonomy-aware feature engineering for microbiome classification," *BMC Bioinformatics,* vol. 19, June 2018.

[12] Q. Zhu, B. Li, T. He, G. Li and X. Jiang, "Robust biomarker discovery for microbiome-wide association studies," *Methods,* vol. 173, p. 44–51, February 2020.

[13] K. Lai, N. Twine, A. O'Brien, Y. Guo and D. Bauer, "Artificial Intelligence and Machine Learning in Bioinformatics," in *Encyclopedia of Bioinformatics and Computational Biology*, Elsevier, 2019, p. 272–286.

[14] N. Pudjihartono, T. Fadason, A. W. Kempa-Liehr and J. M. O'Sullivan, "A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction," *Frontiers in Bioinformatics,* vol. 2, June 2022.

[15] K. Tadist, S. Najah, N. S. Nikolov, F. Mrabti and A. Zahi, "Feature selection methods and genomic big data: a systematic review," *Journal of Big Data,* vol. 6, August 2019.

[16] Y. Saeys, I. Inza and P. Larranaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics,* vol. 23, p. 2507–2517, August 2007.

[17] J. P. Sarkar, I. Saha, A. Sarkar and U. Maulik, "Machine learning integrated ensemble of feature selection methods followed by survival analysis for predicting breast cancer subtype specific miRNA biomarkers," *Computers in Biology and Medicine,* vol. 131, p. 104244, April 2021.

[18] S. S. Verma, A. Lucas, X. Zhang, Y. Veturi, S. Dudek, B. Li, R. Li, R. Urbanowicz, J. H. Moore, D. Kim and M. D. Ritchie, "Collective feature selection to identify crucial epistatic variants," *BioData Mining,* vol. 11, April 2018.

[19] N. LaPierre, C. J.-T. Ju, G. Zhou and W. Wang, "MetaPheno: A critical evaluation of deep learning and machine learning in metagenome-based disease prediction," *Methods,* vol. 166, p. 74–82, August 2019.

[20] B. Bakir-Gungor, H. Hacılar, A. Jabeer, O. U. Nalbantoglu, O. Aran and M. Yousef, "Inflammatory bowel disease biomarkers of human gut microbiota selected via different feature selection methods," *PeerJ,* vol. 10, p. e13205, April 2022.

[21] M. Oh and L. Zhang, "DeepMicro: deep representation learning for disease prediction based on microbiome data," *Scientific Reports,* vol. 10, April 2020.

[22] F. Grazioli, R. Siarheyeu, I. Alqassem, A. Henschel, G. Pileggi and A. Meiser, "Microbiome-based disease prediction with multimodal variational information bottlenecks," *PLOS Computational Biology,* vol. 18, p. e1010050, April 2022.

[23] D. Reiman, A. A. Metwally, J. Sun and Y. Dai, "PopPhy-CNN: A Phylogenetic Tree Embedded Architecture for Convolutional Neural Networks to Predict Host Phenotype From Metagenomic Data," *IEEE Journal of Biomedical and Health Informatics,* vol. 24, p. 2993–3001, October 2020.

[24] Z. Pödör and M. Hekfusz, "Comparing Feature Selection Methods on Metagenomic Data using Random Forest Classifier," Transactions on Engineering and Computing Sciences, vol. 12, pp. 175.-187, 2024.

[25] Y. Zhang, S. Li, T. Wang and Z. Zhang, "Divergence-based feature selection for separate classes," Neurocomputing, vol. 101, p. 32–42, February 2013.

[26] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, J. M. Benítez and F. Herrera, "A review of microarray datasets and applied feature selection methods," Information Sciences, vol. 282, p. 111–135, October 2014.

[27] H. Hacilar, O. U. Nalbantoglu and B. Bakir-Gungor, "Machine Learning Analysis of Inflammatory Bowel Disease-Associated Metagenomics Dataset," in 2018 3rd International Conference on Computer Science and Engineering (UBMK), 2018.

[28] R. J. Urbanowicz, R. S. Olson, P. Schmitt, M. Meeker and J. H. Moore, "Benchmarking relief-based feature selection methods for bioinformatics data mining," Journal of Biomedical Informatics, vol. 85, p. 168–188, September 2018.

[29] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," Computers & Electrical Engineering, vol. 40, p. 16–28, January 2014.

[30] D. He, I. Rish, D. Haws and L. Parida, "MINT: Mutual Information Based Transductive Feature Selection for Genetic Trait Prediction," IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 13, p. 578–583, May 2016.

[31] I. Kavakiotis, P. Samaras, A. Triantafyllidis and I. Vlahavas, "FIFS: A data mining method for informative marker selection in high dimensional population genomic data," Computers in Biology and Medicine, vol. 90, p. 146–154, November 2017.

[32] Y. Shen, J. Xu, Z. Li, Y. Huang, Y. Yuan, J. Wang, M. Zhang, S. Hu and Y. Liang, "Analysis of gut microbiota diversity and auxiliary diagnosis as a biomarker in patients with schizophrenia: A cross-sectional study," Schizophrenia Research, vol. 197, p. 470–477, July 2018.

[33] J. Barrera-Gómez, L. Agier, L. Portengen, M. Chadeau-Hyam, L. Giorgis-Allemand, V. Siroux, O. Robinson, J. Vlaanderen, J. R. González, M. Nieuwenhuijsen, P. Vineis, M. Vrijheid, R. Vermeulen, R. Slama and X. Basagaña, "A systematic comparison of statistical methods to detect interactions in exposome-health associations," Environmental Health, vol. 16, July 2017.

[34] M. Kumar and S. K. Rath, "Classification of microarray using MapReduce based proximal support vector machine classifier," Knowledge-Based Systems, vol. 89, p. 584–602, November 2015.

[35] S. Sasikala, S. A. alias Balamurugan and S. Geetha, "A Novel Feature Selection Technique for Improved Survivability Diagnosis of Breast Cancer," Procedia Computer Science, vol. 50, p. 16–23, 2015.

[36] D. Dernoncourt, B. Hanczar and J.-D. Zucker, "Analysis of feature selection stability on high dimension and small sample data," Computational Statistics & Data Analysis, vol. 71, p. 681–693, March 2014.

[37] M. Jafari, B. Ghavami and V. Sattari, "A hybrid framework for reverse engineering of robust Gene Regulatory Networks," Artificial Intelligence in Medicine, vol. 79, p. 15–27, June 2017.

[38] S. Wang and Y. Cai, "Identification of the functional alteration signatures across different cancer types with support vector machine and feature analysis," Biochimica et Biophysica Acta (BBA) - Molecular Basis of Disease, vol. 1864, p. 2218–2227, June 2018.

[39] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang and H. Liu, "Feature Selection: A Data Perspective," ACM Computing Surveys, vol. 50, p. 1–45, December 2017.

[40] D. M. Farid, A. Nowe and B. Manderick, "A feature grouping method for ensemble clustering of high-dimensional genomic big data," in 2016 Future Technologies Conference (FTC), 2016.

[41] N. Qin, F. Yang, A. Li, E. Prifti, Y. Chen, L. Shao, J. Guo, E. Le Chatelier, J. Yao, L. Wu, J. Zhou, S. Ni, L. Liu, N. Pons, J. M. Batto, S. P. Kennedy, P. Leonard, C. Yuan, W. Ding, Y. Chen, X. Hu, B. Zheng, G. Qian, W. Xu, S. D. Ehrlich, S. Zheng and L. Li, "Alterations of the human gut microbiome in liver cirrhosis," Nature, vol. 513, p. 59–64, July 2014.

[42] J. Qin, R. Li, J. Raes, M. Arumugam, K. S. Burgdorf, C. Manichanh, T. Nielsen, N. Pons, F. Levenez, T. Yamada, D. R. Mende, J. Li, J. Xu, S. Li, D. Li, J. Cao, B. Wang, H. Liang, H. Zheng, Y. Xie, J. Tap, P. Lepage, M. Bertalan, J.-M. Batto, T. Hansen, D. Le Paslier, A. Linneberg, H. B. Nielsen, E. Pelletier, P. Renault, T. Sicheritz-Ponten, K. Turner, H. Zhu, C. Yu, S. Li, M. Jian, Y. Zhou, Y. Li, X. Zhang, S. Li, N. Qin, H. Yang, J. Wang, S. Brunak, J. Doré, F. Guarner, K. Kristiansen, O. Pedersen, J. Parkhill, J. Weissenbach, P. Bork, S. D. Ehrlich and J. Wang, "A human gut microbial gene catalogue established by metagenomic sequencing," Nature, vol. 464, p. 59–65, March 2010.

[43] J. Qin, Y. Li, Z. Cai, S. Li, J. Zhu, F. Zhang, S. Liang, W. Zhang, Y. Guan, D. Shen, Y. Peng, D. Zhang, Z. Jie, W. Wu, Y. Qin, W. Xue, J. Li, L. Han, D. Lu, P. Wu, Y. Dai, X. Sun, Z. Li, A. Tang, S. Zhong, X. Li, W. Chen, R. Xu, M. Wang, Q. Feng, M. Gong, J. Yu, Y. Zhang, M. Zhang, T. Hansen, G. Sanchez, J. Raes, G. Falony, S. Okuda, M. Almeida, E. LeChatelier, P. Renault, N. Pons, J.-M. Batto, Z. Zhang, H. Chen, R. Yang, W. Zheng, S. Li, H. Yang, J. Wang, S. D. Ehrlich, R. Nielsen, O. Pedersen, K. Kristiansen and J. Wang, "A metagenome-wide association study of gut microbiota in type 2 diabetes," Nature, vol. 490, p. 55–60, September 2012.

[44] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and É. Duchesnay, "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[45] A. Kraskov, H. Stögbauer and P. Grassberger, "Estimating mutual information," Physical Review E, vol. 69, p. 066138, June 2004.

[46] H. Peng, F. Long and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, p. 1226–1238, August 2005.

[47] K. Kira and L. A. Rendell, "The Feature Selection Problem: Traditional Methods and a New Algorithm," in Proceedings of the Tenth National Conference on Artificial Intelligence, San, 1992.

[48] R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson and J. H. Moore, "Relief-based feature selection: Introduction and review," Journal of Biomedical Informatics, vol. 85, p. 189–203, September 2018.

[49] L. Breiman, "Random Forests," Machine Learning, vol. 45, p. 5–32, 2001.

[50] M. B. Kursa and W. R. Rudnicki, "Feature Selection with the Boruta Package," Journal of Statistical Software, vol. 36, 2010.

[51] Z. Cai, P. Li, W. Zhu, J. Wei, J. Lu, X. Song, K. Li, S. Li and M. Li, "Metagenomic analysis reveals gut plasmids as diagnosis markers for colorectal cancer," Frontiers in Microbiology, vol. 14, May 2023.

[52] W.-t. Lai, J. Zhao, S.-x. Xu, W.-f. Deng, D. Xu, M.-b. Wang, F.-s. He, Y.-h. Liu, Y.-y. Guo, S.-w. Ye, Q.-f. Yang, Y.-l. Zhang, S. Wang, M.-z. Li, Y.-j. Yang, T.-b. Liu, Z.-m. Tan, X.-h. Xie and H. Rong, "Shotgun metagenomics reveals both taxonomic and tryptophan pathway differences of gut microbiota in bipolar disorder with current major depressive episode patients," Journal of Affective Disorders, vol. 278, p. 311–319, January 2021.

[53] L. J. Marcos-Zambrano, V. M. López-Molina, B. Bakir-Gungor, M. Frohme, K. Karaduzovic-Hadziabdic, T. Klammsteiner, E. Ibrahimi, L. Lahti, T. Loncar-Turukalo, X. Dhamo, A. Simeon, A. Nechyporenko, G. Pio, P. Przymus, A. Sampri, V. Trajkovik, B. Lacruz-Pleguezuelos, O. Aasmets, R. Araujo, I. Anagnostopoulos, Ö. Aydemir, M. Berland, M. L. Calle, M. Ceci, H. Duman, A. Gündoğdu, A. S. Havulinna, K. H. N. Kaka Bra, E. Kalluci, S. Karav, D. Lode, M. B. Lopes, P. May, B. Nap, M. Nedyalkova, I. Paciência, L. Pasic, M. Pujolassos, R. Shigdel, A. Susín, I. Thiele, C.-O. Truică, P. Wilmes, E. Yilmaz, M. Yousef, M. J. Claesson, J. Truu and E. Carrillo de Santa Pau, "A toolbox of machine learning software to support microbiome analysis," Frontiers in Microbiology, vol. 14, November 2023.

[54] A. K. Sharma, S. Bhardwaj, D. K. Srivastava and P. Srivastava, "Type 2 Diabetes Mellitus Prediction with Gut Microbes Using Machine Learning Through Shotgun Metagenomic Sequencing," in Proceedings of World Conference on Information Systems for Business Management, Springer Nature Singapore, 2024, p. 21–32.

[55] J. Shen, A. G. McFarland, R. A. Blaustein, L. J. Rose, K. A. Perry-Dow, A. A. Moghadam, M. K. Hayden, V. B. Young and E. M. Hartmann, "An improved workflow for accurate and robust healthcare environmental surveillance using metagenomics," Microbiome, vol. 10, December 2022.

[56] F. Degenhardt, S. Seifert and S. Szymczak, "Evaluation of variable selection methods for random forests and omics data sets," Briefings in Bioinformatics, vol. 20, p. 492–503, October 2017.

[57] Y. Gao, Z. Zhu and F. Sun, "Increasing prediction performance of colorectal cancer disease status using random forests classification based on metagenomic shotgun sequencing data," Synthetic and Systems Biotechnology, vol. 7, p. 574–585, March 2022.

[58] T. Loganathan and G. Priya Doss C, "The influence of machine learning technologies in gut microbiome research and cancer studies - A review," Life Sciences, vol. 311, p. 121118, December 2022.

[59] P. Tonkovic, S. Kalajdziski, E. Zdravevski, P. Lameski, R. Corizzo, I. M. Pires, N. M. Garcia, T. Loncar-Turukalo and V. Trajkovik, "Literature on Applied Machine Learning in Metagenomic Classification: A Scoping Review," Biology, vol. 9, p. 453, December 2020.

[60] M. Ziemski, T. Wisanwanichthan, N. A. Bokulich and B. D. Kaehler, "Beating Naive Bayes at Taxonomic Classification of 16S rRNA Gene Sequences," Frontiers in Microbiology, vol. 12, June 2021.

[61] X.-W. Wang and Y.-Y. Liu, "Comparative study of classifiers for human microbiome data," Medicine in Microecology, vol. 4, p. 100013, June 2020.

[62] C. Bentéjac, A. Csörgő and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," Artificial Intelligence Review, vol. 54, p. 1937–1967, August 2020.

[63] T. H. Nguyen, E. Prifti, Y. Chevaleyre, N. Sokolovska and J.-D. Zucker, Disease Classification in Metagenomics with 2D Embeddings and Deep Learning, arXiv, 2018.

[64] M. A. Rahman and H. Rangwala, "RegMIL: Phenotype Classification from Metagenomic Data," in Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, 2018.

[65] M. Bhat, B. M. Arendt, V. Bhat, E. L. Renner, A. Humar and J. P. Allard, "Implication of the intestinal microbiome in complications of cirrhosis," World Journal of Hepatology, vol. 8, p. 1128, 2016.

[66] X.-y. Huang, Y.-h. Zhang, S.-y. Yi, L. Lei, T. Ma, R. Huang, L. Yang, Z.-m. Li and D. Zhang, "Potential contribution of the gut microbiota to the development of portal vein thrombosis in liver cirrhosis," Frontiers in Microbiology, vol. 14, October 2023.

[67] X. C. Morgan, T. L. Tickle, H. Sokol, D. Gevers, K. L. Devaney, D. V. Ward, J. A. Reyes, S. A. Shah, N. LeLeiko, S. B. Snapper, A. Bousvaros, J. Korzenik, B. E. Sands, R. J. Xavier and C. Huttenhower, "Dysfunction of the intestinal microbiome in inflammatory bowel disease and treatment," Genome Biology, vol. 13, p. R79, 2012.

[68] S. W. Ryu, J.-S. Kim, B. S. Oh, W. J. Choi, S. Y. Yu, J. E. Bak, S.-H. Park, S. W. Kang, J. Lee, W. Y. Jung, J.-S. Lee and J. H. Lee, "Gut Microbiota Eubacterium callanderi Exerts Anti-Colorectal Cancer Activity," Microbiology Spectrum, vol. 10, December 2022.

[69] A. P. Doumatey, A. Adeyemo, J. Zhou, L. Lei, S. N. Adebamowo, C. Adebamowo and C. N. Rotimi, "Gut Microbiome Profiles Are Associated With Type 2 Diabetes in Urban Africans," Frontiers in Cellular and Infection Microbiology, vol. 10, February 2020.

[70] Y.-H. Xie, Q.-Y. Gao, G.-X. Cai, X.-M. Sun, T.-H. Zou, H.-M. Chen, S.-Y. Yu, Y.-W. Qiu, W.-Q. Gu, X.-Y. Chen, Y. Cui, D. Sun, Z.-J. Liu, S.-J. Cai, J. Xu, Y.-X. Chen and J.-Y. Fang, "Fecal Clostridium symbiosum for Noninvasive Detection of Early and Advanced Colorectal Cancer: Test and Validation Studies," EBioMedicine, vol. 25, p. 32–40, November 2017.

[71] M. A. Osman, H.-m. Neoh, N.-S. Ab Mutalib, S.-F. Chin, L. Mazlan, R. A. Raja Ali, A. D. Zakaria, C. S. Ngiu, M. Y. Ang and R. Jamal, "Parvimonas micra, Peptostreptococcus stomatis, Fusobacterium nucleatum and Akkermansia muciniphila as a four-bacteria biomarker panel of colorectal cancer," Scientific Reports, vol. 11, February 2021.

[72] X. Liu, Y.-W. Cheng, L. Shao, S.-H. Sun, J. Wu, Q.-H. Song, H.-S. Zou and Z.-X. Ling, "Gut microbiota dysbiosis in Chinese children with type 1 diabetes mellitus: An observational study," World Journal of Gastroenterology, vol. 27, p. 2394–2414, May 2021.

[73] A. Le Goallec, B. T. Tierney, J. M. Luber, E. M. Cofer, A. D. Kostic and C. J. Patel, "A systematic machine learning and data type comparison yields metagenomic predictors of infant age, sex, breastfeeding, antibiotic usage, country of origin, and delivery type," PLOS Computational Biology, vol. 16, p. e1007895, May 2020.