Investigating ensemble methods for essential gene predictions in bacteria

Vanisha Patel

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Philosophy

Original Submission: October 2020
Second Submission: October 2021

Acknowledgements

I will be forever grateful to my supervisor Jamie Twycross for both allowing me to experience a whole new discipline and his support during my research. In particular, I would like to thank him for his patience and understanding during a very difficult period of my life. His advice and encouragement regarding my future has allowed me to close this chapter of my life with confidence.

Thanks to the wonderful team at the SBRC for all their help and the BBSRC and School of Computer Science for funding my project.

A huge heartfelt thank you goes out to my fellow inhabitants of B18: Rupert Norman, Nicole Pearcy, James Gilbert, Mohit Dalwadi, Thomas Millat and Claudio Andrino. For many ridiculous memories and years of laughing till my sides hurt.

I would like to thank Scott Jones, Ben Underwood and Shona Patel for their valiant attempts to keep me sane, and also the rest of my family and friends for their support. Especially my parents for providing the foundations on which I have been able to build my future.

Most importantly, I would like to thank Simon Cobley for his unwavering love and encouragement in everything I do, and for also being the reason I have Eva and Cleo.

Abstract

Essential genes are the genes required for an organism to survive in stable conditions with an abundance of nutrients. The identification of essential genes is important to both our understanding of bacterial organisms and our ability to manipulate them. Many machine learning methods have been proposed for the prediction of essential genes. However, the majority of these studies have a limited focus, i.e. a single optimised classifier and feature set combination to predict genes within the same organism. Therefore, as the models have a narrow scope they cannot be reliably applied to newly sequenced organisms. This ability of a model to generalise to new data can be improved by increasing the dataset and combining results from different classifiers.

The aim of this thesis was to develop an ensemble method to predict essential genes in bacteria. In total 62 commonly used sequence based features and 7 supervised learning classifiers were identified from the literature. Using online databases, 73 studies with high quality laboratory essentiality data were collated for 45 bacterial strains. To build the ensemble base learners, feature selection algorithms were used to generate feature subsets. Analysis of the subsets showed that while particular features were selected more frequently by the algorithms, no features were completely excluded. The performance of each subset with the classifiers was investigated to identify feature sets for the ensemble base learners.

Through studying the performance of the feature sets as part of a majority voting ensemble algorithm, we were able to show that for cross validation the ensemble approach performance was higher than the individual classifiers. This was confirmed through validation testing on organism with no matching genus in training data.

The results show that it is possible to improve the ability of a classifier to generalise to new organisms through the application of feature selection and ensemble learning.

List of Figures

List of Tables

Abbreviations

A	alanine
A3s	nucleotide A at silent site
ACID	acidic amino acids
ALIP	aliphatic amino acids
ARO / AROM	aromatic amino acids
AUC	area under the ROC curve
avAUC	average area under the curve
BASIC	basic amino acids
C	cysteine
C3s	nucleotide C at silent site
CAI	codon adaptation index
CBI	codon bias index
CELL	cellwall
CFS	correlation-based feature selection
CHAR	charged amino acids
CV	cross validation
CYT	cytoplasm
CYTM	cytoplasmic membrane
D	aspartic acid
DEG	database of essential genes
DNA	deoxyribonucleic acid
DT	decision tree
E	glutamic acid
EXT	extracellular
F	phenylalanine
FBA	flux balance analysis
Fop	frequency of optimal codons
G	glycine
G3s	nucleotide G at silent site
GC	nucleotides G + C in gene
GC3s	nucleotides G + C at silent sites
H	histidine
HYD	hydrophobicity of protein
I	isoleucine
ISO	isoelectric point
K	lysine
kNN	k-nearest neighbour

L	leucine
L_aa	length of protein
LASSO	least absolute shrinkage and selection operator
LIN	support vector machine with linear kernel
LR	logistic regression
M	methionine
mRNA	messenger ribonucleic acid
MW	molecular weight
N	asparagine
NB	naive bayes
NCBI	national center for biotechnology information database
NN	neural network
NONP	non-polar amino acids
OGEE	online gene essentiality database
OUTM	outer membrane
P	proline
P-3, P-5, P-7, P-10, P-15, P-20, P-30	paralogs at e-value
PERI	periplasm
POL	polar amino acids
PPI	protein–protein interactions
Q	glutamine
R	arginine
RARE	rare amino acids
RF	random forest
RFE	recursive feature elimination
ROC	receiver operating characteristic
S	serine
SBS	sequential backward selection
SFS	sequential forward selection
SMALL	small amino acids
STRAND	genome strand
SVM	support vector machine
T	threonine
T3s	nucleotide T at silent site
TINY	tiny amino acids
TM60	amino acids in helices in first 60 nucleotides
TM60H	transmembrane helices in first 60 nucleotides
TMAA	amino acids in helices
TMH	transmembrane helices
V	valine
VAR	variance
W	tryptophan
Y	tyrosine

Contents

Chapter 1

Introduction

1.1 Introduction

This introduction chapter outlines the motivation for the project along with the aim of the thesis and the research questions we set out to investigate. The chapter concludes with a brief summary of each subsequent chapter.

While all biological terms are briefly outlined in the text, to aid understanding for a non-biologist, after the chapters is included a glossary where the terms are described in the context of this thesis. This is provided because a general understanding of the terms is important for the discussions.

1.2 Background and motivation

Synthetic biology is a field which focuses on the re-engineering of microorganisms for useful purposes. It encompasses both the design and construction of new biological systems, and the re-designing of existing natural systems [1,2]. As a multi-disciplinary field with boundless potential, it has been applied to the development of everything from drug delivery systems [3–5], to living machines (biobots) [6–8], and the essentials of human space exploration [9, 10].

The rapid growth of synthetic biology is, in part, due to the push for petrochemical replacements. Our modern life is completely dependent on the products derived from non-renewable sources such as crude oil, natural gas and coal, but as our consumption increases, so to does the strain on the supply chain and environment [11]. Finding greener and sustainable alternatives of microbial origin has become an important area for use in industrial biotechnology [12–17].

The "perfect" organism for use in chemical production is one whose

behaviour and products can be accurately anticipated and controlled [18, 19]. The ideal way to achieve this is by having an organism with only the minimum number of genes needed for it to survive (a minimal genome), and then adding the genes required for it to produce the desired chemical for production [20, 21]. Identifying the genes for survival (essential genes) has become the cornerstone of synthetic biology [22, 23].

The importance of essential genes and the minimal genome concept has accelerated the advancement of experimental methods within biology which has, in turn, reduced the cost of genome sequencing [24]. The advantage of this is that there are now more genome and essentiality data available for use in computational methods. As a result the application of machine learning to biology has been steadily increasing. While many different pipelines and algorithms are being applied to the important task of gene essentiality prediction [24–27]. Majority of methods are highly specific, being targeted at predictions within the same species or closely related organisms. Or they require additional information such as gene function, this information is available for model organisms and while it can be predicted for unknown genes. The accuracy of this prediction depends relationship between the organisms.

1.3 Aims and objectives

The aim of this project is to develop an ensemble method for predicting essential genes in bacteria using multiple classifiers and feature sets. In order to achieve this a number of key objectives need to be met:

1. Investigate literature to identify classifiers and features commonly applied to the computational identification of essential genes.

2. Collect existing gene essentiality data that can be used in supervised machine learning.

3. Investigate, for the identified classifiers, which features or sets of features work best for essential gene prediction.

4. Develop an ensemble method using the identified feature subsets and classifiers.

5. Assess the developed ensemble method on the datasets identified, including an evaluation of prediction accuracy in comparison to state-of-the-art gene prediction methods.

1.4 Research questions

The aims and objectives relate to the specific research questions, outlined below, to be addressed within this thesis.

- **How is gene essentiality defined?**
  The definition of an essential gene is based entirely on the biological impact it has on the organism. It is an important concept in all computational prediction methods as it affects the data selected to train machine learning models.

- **What machine learning models currently exist to predict gene essentiality?**
  As the diversity of base learners is an important factor for constructing ensembles, identifying the commonly utilised machine learning classifiers for computational gene prediction is key for construction.

- **How does feature selection impact classifier performance?**
  The prediction accuracy of a classifier largely depends on the data it is trained with. This requires an investigation into how different subsets of features perform with the classifiers. Knowing how feature subsets and classifier combinations perform provides a base for building an ensemble model.

- **How well does the developed ensemble method predict gene essentiality compared to a single classifier approach?**
  After the construction of novel ensemble models it is important to investigate whether they perform better than the individual parts. Two performance methods commonly applied are cross validation and unseen test data.

1.5 Organization of the thesis

This section provides a short outline for each chapter of this thesis.

Reviewed in Chapter 2 is the literature relevant for this thesis. It contains an overview of bacterial gene essentiality and the limitations of laboratory experiments that produce the data used in computational techniques. The rest of the chapter covers the important machine learning approaches on which our project is based. Described in chapter 3 is the gene essentiality data collected for use with the machine learning algorithms. This was done

from two online sources the Database of Essential Genes (DEG) and the Online Gene Essentiality database (OGEE).

Chapter 4 begins by describing the common features adopted from the literature and how they were generated. It then covers the application of the feature selection algorithms and the subsequent feature subsets created for the DEG dataset. The performance of each subset with the machine learning classifiers is then analysed and discussed.

Using the feature subsets from Chapter 4, Chapter 5 details the building of the ensemble model using them and assesses the models prediction accuracy through both cross validation and using OGEE as unseen data. Included is an evaluation of the predictions generated by the ensemble model and a comparison to state-of-the-art methods in gene prediction.

Finally, the thesis is concluded in Chapter 6 which provides a discussion of how well the aims and objectives were addressed and possible future directions worth pursuing. A brief outline of the project designed to address the research questions is shown in Figure 1.1.

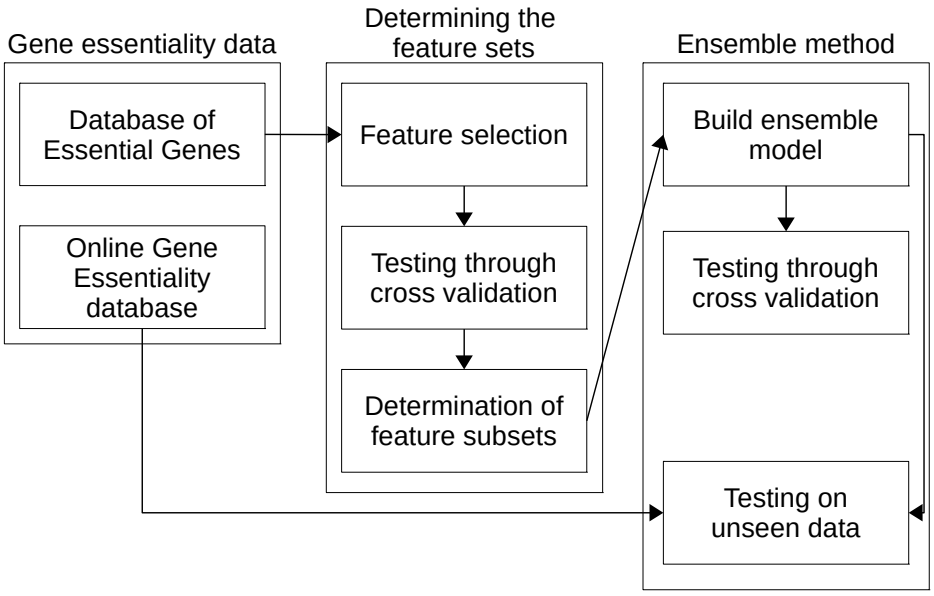

Figure 1.1: **An overview of the project.** An overall summary of the project in this thesis and how the parts link together.

Chapter 2

Literature Review

2.1 Introduction

This chapter reviews the literature related to the project. Firstly, an overview of bacterial essential genes and their limitations in synthetic biology. Secondly, the key points regarding the laboratory experiments that produce the data utilised by computational techniques. Finally, a description of machine-learning based approaches to essential gene prediction. Also provided is a summary of key studies on which the project is based.

2.2 What is an essential gene?

The genome of an organism contains a complex mass of genes which allow it to thrive in many different environments. Contained within this mass of genes are those we consider to be essential for survival. The deletion or disruption of any one of these genes is lethal.

In this study we consider essential genes as those coding for functions vital for survival in conditions free from environmental stress and containing all necessary nutrients [28]. These genes are also referred to as minimal genes.

The term minimal gene set refers to this smallest possible group of essential genes required for organism survival. Any genes not in the minimal gene set are considered to be non-essential. There are two main online sources which contain information about these minimal gene sets in a number of different organisms, the database of essential genes [29–31] and the online gene essentiality database [32,33]. For synthetic biology, being able to identify and then disrupt or modify these genes can help remove any undesirable interactions and make biological systems more predictable. Hence why the

hunt for essential genes has become one of the cornerstones of the field. The search is also of high importance in the medical field as genes essential for growth in a media where all the nutrients are provided, such as our bodies, make good targets for broad-spectrum antibacterial drugs [34].

One of the problems in understanding gene essentiality is the vast difference in genome size between organisms. One of the smallest genomes belonging to *Mycoplasma genitalium* contains only 482 genes [35], whereas *Staphylococcus aureus*, MRSA, contains ∼3000 genes. While the number of essential genes in both ranges at 350 genes, the increase in non-essential genes complicates the relationships within the organism and which in turn increases the difficult in study.

The most efficient way to achieve high yield production is through altering an organism which naturally produces the target product. Genes can be removed from the organism to limit the production of unwanted side products and interactions, while also increasing the target yield. But for an organism to contain a pathway for an unusual product it will generally have lots of other adaptations and therefore a large genome.

2.2.1 Different types of essentiality

Two key functions of an organisms survival depend on its ability to metabolise and reproduce. So most essential genes relate to these functions. While we have adopted the minimal gene definition of an essential gene, the term "essential gene" is open to debate because essentiality ultimately depends on the conditions in which the organism is growing. There are some other groups and definitions of essential genes, they are broadly grouped into the following types:

Conditionally essential These genes are essential under specific growth conditions. For example, genes required for survival in presence of a toxin or to utilise different food sources.

Essential for fitness This category contains genes which if removed result in a measurable decrease in the growth rate or metabolism of the organism but does not result in death [36]. They are hard to identify as slow growth can sometimes be classed as cell death which results in mis-classified genes.

Synthetic lethal genes These are combinations of genes in which the inactivation of two or more of them result in death [37, 38]. The most

common reason for synthetic lethality is due to the duplication of genes in the genome, called paralogs. Genomes normally contain paralogs, making synthetic lethals one of the most important things to be considered in the transferring of computational predictions to a laboratory experiment. One option to overcome this is to identify essential genes and then also have a list of paralog genes that are potentially essential.

2.3 *In vivo* prediction methods

Over the years a range of wet-laboratory techniques have been applied to the investigation of gene essentiality. The methods consist of either randomly or systematically inactivating genes and their essentiality is determined depending on whether the organism survives [39]. Currently in use are three main types of biological techniques: transposon mutagenesis, gene knockouts, mRNA interface.

The most popular whole genome approach is transposon mutagenesis and a large portion of the data collected from online databases utilise this approach. There are some key drawbacks to this method which result in mis-classified genes, these usually include smaller genes which are rarely disrupted or an insertion at either end of a gene which may not be sufficient to disrupt expression [30, 36].

While the efficiency of wet-laboratory techniques has greatly improved over the years, they remain time-consuming and expensive tasks. These problems apply more so to the study of non-model organisms and those that are hard to cultivate and manipulate. Scenarios include those with unusual products of interest but which have large genomes with many interactions.

The advancement of laboratory techniques has resulted in the production of vast amounts of data. This has aided the development of computational based methods which enable faster identification of candidate essential genes, with ever increasing accuracy. Computational methods can circumvent some of the laboratory limitations for non-model organisms by providing the necessary information to speed up experiments.

2.4 Machine learning approaches

With the ability to faster identify essential candidates, computational methods provide an appealing alternative to laboratory experiments as they circumvent the expensive and difficult experimental screens.

Many different computational methods have been applied to the search for essential genes. But as the prediction of essential genes or proteins can be defined as a classification problem, machine learning approaches can be applied. There is an ever growing number of classification algorithms available and they can be grouped by their style of learning, i.e. supervised learning, unsupervised learning or semi-supervised learning. For determining gene essentiality, supervised learning is applied because the labels or classes of the training data are known.

Supervised learners are able to classify genes by constructing and training a classifier using features or attributes associated with the classes (essential or non-essential genes) [40,41]. The classifier is trained using the known essentialities of well-studied genomes and then used to predict essentiality in another genome [42–44].

There are many different classification algorithms available for supervised learning. But as thei r performance depends on the specific problem to which they are being applied, it is not possible to conclude that one algorithm is superior to another. The 'perfect' classifier for a problem depends on the quality of the training data, the features selected and the unclassified data [45]. One assumption of supervised learning is that the training and testing data have the same distribution. While, organisms that are closely related naturally have this feature, as distance between them increases the distribution is more likely to be different [41].

As the quality of predictions depends on the relationship between the training and testing data, machine learning is not currently suitable for predicting conditional essential genes. The genes that are essential to an organism are highly dependent on the environmental conditions. A classifier trained with genes under stable conditions will not have the same class attributes for genes under other conditions and therefore not accurately classify genes. As computational techniques are becoming popular there may soon be enough data under the same condition to use for training a classifier.

Classifiers

The application of classifiers to the prediction of essential genes and proteins has already been extensively reviewed in a number of papers [26, 46–49]. While some studies reviewed focus on the implementation of novel and highly optimised algorithms, existing classifiers are also implemented. The most common of which are decision trees [40,44,50], k-nearest neighbor [51], logistic regression [40, 52], naive bayes [40, 41, 53], neural networks [42],

random forest [52, 52] and support vector machines [44, 51, 54–56].

Features

A variety of features have been associated with gene and protein essentiality, and therefore have been applied in machine learning predictions. As essential genes are under unique evolutionary pressure, it is likely that they share many other characteristics which may be gleaned from genome sequence data. They can be broadly classified into two main groups, sequence derived features and context dependent features [57, 58].

Sequence derived features This group contains features which can be generated directly from either the DNA or protein sequences. These features can be further split into: sequence information, homology, physicochemical properties and subcellular localisation.

Sequence information includes features such as the GC content of a gene, a high GC content is thought to be more robust and stable [59]. Codon usage in essential genes is more rigid than non-essential genes [60]. Strand bias as essential genes tend to be encoded on the leading strand of the chromosome [61, 62]. Protein length, where essential genes are usually on the higher or lower end regarding length [63, 64].

Homology based features include paralogs. Essential genes should have a fewer number of paralogs compared to non-essential genes [65].

Cellular localization of proteins have been shown to be important features in the identification of essential genes, with nuclear localisation demonstrating the strongest positive correlation with essentiality [59, 66]. Other features (areas of localisation) include the cytoplasm, outer membrane and external.

Context dependent features The features in this group are generated from experimental data involving the study of both genes and proteins. The features can be grouped further into network topology and gene expression.

Network topology based features require the availability or construction of protein-protein interaction (PPI) networks, gene regulatory networks or metabolic networks. From these networks features are generated. In PPI networks the genes or proteins are connected and the essential genes tend to be more highly connected than non-essential genes. Common features of networks are degree centrality, betweenness centrality, closeness centrality and subgraph centrality [57, 66–68]. Network topology based features are possible because of whole genome expression studies. Fang *et al.* reviews the

use of PPI networks in combination with other features in the application of essential genes [48].

Gene expression features are based on mRNA expression levels, how much the gene is being expressed, and fluctuations in gene expression. The higher and more stable expression levels are for a gene, the more likely it is to be essential [69].

While context dependent features are often used in prediction studies as experimental and genetic network information is available for the well-studied species, the features are not available for new and under-studied organisms limiting the application of the prediction model created. As whole genome sequencing is now widely accessible, sequence derived features have a more generalizable application and can achieve highly accurate predictions [39, 59, 70].

Feature selection

Compared with the traditional homology mapping, supervised machine learning methods use more features to make predictions. Over the years a wide range of features have been linked with gene essentiality. However, increasing the number of features causes an increase in the dimension of the feature space, resulting in a dramatic increase of computational complexity. Here feature selection is used to reduce the feature set making it a key process in machine learning [40, 71].

As well as the increase in computational complexity, it has been shown that having a large number of features is not optimal for classification as many of the features may be redundant and can reduce the accuracy of the predictions. By reducing the feature set we aim to avoid overfitting and therefore improve model performance on unseen data while simultaneously producing more cost-effective models.

There are two main methods for reducing the feature set, the first being dimension reduction where new features are created by combining the original features, these types of approaches include principal component analysis or information theory. The second method being feature selection where the most relevant features are selected to create a subset of the original. Here, as the original features are not altered, it has the advantage of allowing interpretability by a domain expert.

Feature selection techniques differ in the way they search for feature subsets and the way they interact with the classifier. For classification the techniques can be split into three categories: filter methods, wrapper

methods and embedded methods.

Filter methods are the simplest and fastest, they select feature subsets based on the relationship between the feature and the class. These types of methods include variance. Advantages of these methods are they they can easily be scaled to high-dimensional datasets, have a high number of features, they are computationally simple and fast, while also independent of the classification algorithm. Feature selection is only carried out once before being evaluated on the classifiers. However, as each feature is considered separately feature dependencies are ignored, which may lead to worse classification performance when compared to other types of feature selection techniques.

Wrapper methods search for the best performing subset by taking each possible subset, one-by-one, and training and testing the classifier. These methods include reverse feature elimination. Advantages include the interaction with classifier to choose more meaningful feature subsets and the ability to take into account feature dependencies. A disadvantage of these methods are that they have a higher risk of overfitting than the filter methods and are computationally expensive as the number of features grows and this applies more so if the classifier is also computationally expensive.

In embedded methods, also referred to as intrinsic methods, feature selection is done as part of the classifier construction. Examples include decision trees. Advantages include the interaction with the classification model, while at the same time being far less computationally intensive than wrapper methods.

While feature selection has many advantages, the search for a subset introduces an additional layer of complexity in the modelling task and as the performance and generalization of classifiers is directly affected by feature selection, it is no small task.

2.4.1 Ensemble approaches

Ensemble based learning is a general term for approaches that combine multiple classifiers and whose decisions are combined to improve the performance of the overall system. In the literature they are also referred to as multiple classifiers, a committee of classifiers or a mixture of experts. The individual classifiers in an ensemble are called base-learners or inducers and can be any machine learning algorithm.

It is widely known that through combining multiple classifiers it is possible to achieve more accurate predictions compared to using single classifiers.

This is due to the idea that an error in a single inducer will be compensated for by the other classifiers and as a result the overall prediction performance of an ensemble is better than that of a single inducer [72, 73]. As a result, ensemble approaches take advantage of the differences in the base learners to improve accuracy and reliability.

The results from the base learners can be combined in a number of ways. The most common method applied in gene studies is majority voting, this is where the the output is the most represented class among the base learners. A majority voting system can be optimised through assigning different weights to the result of each base learner [74, 75].

In the area of gene and protein studies ensemble methods have been applied in range of studies, including to the prediction of protein subcellular locations [76] and protein structural classes [77]. One of the first applications of ensemble methods to the prediction of essential genes was in 2006 when Seringhaus *et al.* [59] trained a hybrid system that combined the outputs of decision trees, naive bayes and logistic regression models. The system was trained on *Saccharomyces cerevisiae* and used to predict essential genes in *Saccharomyces mikatae*.

2.4.2 Summary of the literature

Table 2.1 on the following pages contains the key studies from the literature on which the project is based. For each study is listed the specific aspects that were selected for the project in regards to features, feature selection algorithms and classifiers.

Table 2.1: Key studies on which the thesis is based.

Features	**Classifiers**	**Number of organisms**	**References**
Sequence Paralogs Physiochemical Subcellular	SVM Neural Network Decision Tree	14	Palaniappan et al., 2011 [78]
Sequence Paralogs Physiochemical Subcellular	Logistic Regression	2	Deng et al., 2012 [79]
Sequence Paralogs Physiochemical Subcellular	SVM	31	Liu et al., 2017 [80]
Sequence Physiochemical Protein domain PPI network	PCR	2	Lin et al., 2017 [81]
Sequence Paralogs Physiochemical Subcellular Protein domain GCE network	Ensemble: Naive Bayes Logistic Regression Decision Tree CN2 Rule	2	Deng et al., 2011 [40]
Sequence Paralogs PPI network Conservation	Separate & Ensemble Weighted kNN SVM	1	Saha et al., 2006 [51]
Sequence Physiochemical	SVM	16	Ning et al., 2014 [56]
Sequence Physiochemical Subcellular	Ensemble: Decision Trees Random Forest Logistic Regression zeroR Rule Naive Bayes	2	Seringhaus et al., 2006 [59]
Sequence Paralogs PPI network	Naive Bayes	21	Cheng et al., 2014 [41]

Continued on next page

Table 2.1 : continued from previous page

Features	Classifiers	Number of organisms	References
Sequence Paralogs PPI network GCE network	SVM Neural Network	1	Chen et al., 2005 [42]
Sequence Paralogs Subcellular PPI network	Naive Bayes	2	Gustafson et al., 2006 [70]

2.5 Other *in silico* prediction methods

Other computational methods have also been applied to the prediction of essential genes. Mushegian and Koonin [82] were the first to develop a computational method in 1996 based on comparative genomics, also commonly referred to as homology mapping or sequence homology. Homology refers to genes descended from a common ancestral DNA sequence. The technique consists of comparing the sequences of genes with known essentiality against the genome of interest. Sequences that are above a defined percentage match and coverage length are grouped as homologs. Essential genes are often more conserved than nonessential genes in the process of long-term evolution and should therefore be present in most bacteria [58]. This method limits the search to conserved genes between organisms which often accounts only for a small portion of the genome [83], it also overlooks highly evolving genes leading overall to an underestimation of essential genes [84].

Protein network topology models are based on the physical interactions of two or more proteins, protein–protein interactions (PPIs). These interactions can be studied in large-scale genome studies and the data used to build networks in which the essentiality of a gene can be studied. Proteins in the network which are highly linked to their neighbours tend to be essential [85]. Various attributes of the network nodes are used as features in classifiers [51, 70, 86].

Genome-scale metabolic networks allow us to study the interactions between proteins in terms of their metabolism [87,88]. A common constrained-based approach is flux balance analysis (FBA) [88, 89]. FBA has been used

to simulate gene knockouts and the lethality on the network, therefore identifying essential genes [89].

These methods have been reviewed for their application to gene essentiality in [47, 48]. Along with less commonly applied methods, including different types of networks [50, 66] and orthologous properties [90].

2.6 Chapter summary

This chapter reviewed the importance of bacterial essential genes, the biological definitions on which studies can be based and a brief outline of the common laboratory methods used to generate the data utilised in computational studies. Also covered are some of the limitations and assumptions that need to be carried forward into computational methods. Most current computational methods use small subsets of closely related organisms or require the use of complicated features, as they accuracy of these predictions depend on the organisms being closely related, their accuracy decreases when applied to non-model or unknown organisms. The chapter then focuses on machine learning approaches, reviews the different aspects that affect their performance and how they have been approached in previous studies. Finally, a summary table of the studies on which this project was based is provided.

Chapter 3

Gene Essentiality Data

3.1 Introduction

The performance of a machine learning approach depends on the availability of high quality data. The data used for training needs to be accurate, complete and have minimal noise [91]. This chapter outlines the databases and details for the essential gene data gathered.

3.2 Method

Gene essentiality information was downloaded from two online databases, the Database of Essential Genes [29–31] and the Online Gene Essentiality database [32,33], they are described in the following sections. Although the genes in the databases contain essentiality labels, they lack other useful information about the gene such as which strand it is located on or its position in the genome. For this reason the essentiality information obtained from the databases was combined with the National Center for Biotechnology Information database (NCBI) [92], which contains more detailed information about the genes, but does not contain essentiality information.

For each bacterial species identified from the essentiality databases the corresponding genbank (ftp://ftp.ncbi.nlm.nih.gov/genbank) [93] file was downloaded from NCBI. For every bacterial gene, its gene identifier was used to find a match within the genbank file. Where a match could not be found the gene identifier was instead used to obtain a protein identifier from NCBI. This protein identifier was then used to find a match with the genbank file. If the protein identifier also resulted in no match the gene was excluded from the final dataset.

There were some genes for which no matches could be found. The reason

for this being that the genbank files in NCBI are constantly being updated with results from new experiments and this causes the gene identifiers in the files to change. However the online databases do not update at the same rate as NCBI and therefore the genes from the online databases can refer to older experiments.

Some organisms in the online databases were not included in our study. For these organisms, gene essentiality was determined for different conditions, such as the presence of the antibiotic Tobramycin or for bile acid tolerance. Essentiality for all other organisms used is based on rich conditions, meaning only the genes needed for basic functions should be active. Details about the specific organisms excluded from our study are included in the separate database sections to follow. At the end of the labeling a dataset containing: nucleotide gene sequences, gene or protein identifier in the genbank file and an essentiality label was produced.

3.3 Database of essential genes

Constructed in 2003 by Zhang and his colleagues the Database of Essential Genes, referred to as DEG, contained all the essential genes available at the time, along with the ability to run a simple comparison between a query sequence and those in DEG [29]. As the essential gene field progressed DEG similarly followed suit and by 2008 it had reached its fifth iteration. There was a significant increase in the eukaryote genes contained, but a 10-fold jump in prokaryotic essential genes pushing the number to over 5,000, for both some of the genes initially collected from the literature and those determined by theoretical predictions were replaced by genes determined through genome-wide studies [31].

Along with an increase in genes DEG 10 saw the addition of essential non-coding elements, customisable BLAST tools that allow species- and experiment-specific searches for either single genes or a list, annotated and unannotated genomes [30]. Within DEG are also sub-databases which contain non-essential coding genes, these can be inferred from the set of essential genes or based on the original source. This information can come from transposon mutagenesis studies which determine non-essential genes first while the essential genes are inferred.

Version 15.2, used in this study, sees DEG divided into four main sections: non-coding; archaea; bacteria and eukaryotes, all except non-coding can be searched for both DNA and amino acid sequences. It now contains over 20,000 essential genes and 600 essential non-coding sequences across

the database, with 36 different organisms within bacteria [30].

For this study 40 bacterial species were selected from DEG15.2, as their essentialities are based on growth in rich media (the presence of all vital nutrients). Information on the organisms is shown in Table 3.1. For the species selected, the genes obtained from DEG were matched to a genbank file using the method described in Section 3.2. Of the 40 bacterial species 29 are gram negative and 11 gram positive organisms and 15,043 essential genes and 93,808 non-essential genes.

The data for seven organisms was excluded from the final dataset for two main reasons. The first being that the conditions under which essentiality was determined did not match our definition of essential as discussed in a previous chapter, the second reason was that after the labeling step there were no matched genes for these organisms. A list of the organisms and reasons for exclusion is included in Appendix A.

Table 3.1: **Information on the 40 bacterial species collected from DEG.** Shown for each organism: whether it is gram-negative (-) or gram-positive (+), the number of essential genes in DEG and the number of essential and non-essential genes in the final dataset after being matched to Genbank files.

Organism name	**Gram-positive (+) / Gram-negative (-)**	**Essential genes in dataset**	**Total genes in dataset**
Acinetobacter baumannii ATCC 17978	-	458	3351
Acinetobacter baylyi ADP1	-	499	2332
Agrobacterium fabrum C58	-	361	5154
Bacillus subtilis 168	+	271	4175
Bacillus thuringiensis BMB171	+	516	449
Bacteroides fragilis 638R	-	547	3625
Bacteroides thetaiotaomicron VPI-5482	-	325	4778
Brevundimonas subvibrioides ATCC 15264	-	412	2780
Burkholderia thailandensis E264	-	406	3873
Burkholderia pseudomallei K96243	-	505	5726
Campylobacter jejuni subsp. jejuni NCTC 11168 = ATCC 700819	-	166	1572
Campylobacter jejuni subsp. jejuni NCTC 11168 = ATCC 700819	-	228	1572
Caulobacter crescentus	-	480	3051
Escherichia coli MG1655 I	-	609	3523
Escherichia coli MG1655 II	-	296	4357

Continued on next page

Table 3.1 : continued from previous page

Organism name	Gram-positive (+) / Gram-negative (-)	Essential genes in dataset	Total genes in dataset
Escherichia coli ST131 strain EC958	-	315	4639
Francisella novicida U112	-	392	1572
Haemophilus influenzae Rd KW20	-	642	1128
Helicobacter pylori 26695	-	323	1349
Mycobacterium tuberculosis H37Rv III	+	687	3626
Mycobacterium tuberculosis H37Rv	+	614	3085
Mycoplasma genitalium G37	+	381	437
Mycoplasma pulmonis UAB CTIP	+	310	489
Porphyromonas gingivalis ATCC 33277	-	463	1564
Porphyromonas gingivalis ATCC 33277	-	281	1564
Pseudomonas aeruginosa PAO1	-	336	5515
Pseudomonas aeruginosa UCBPP-PA14	-	335	1164
Salmonella enterica serovar Typhi Ty2	-	358	3940
Salmonella enterica serovar Typhimurium SL1344	-	353	3893
Salmonella typhimurium LT2	-	230	4449
Shewanella oneidensis MR-1	-	402	1505
Sphingomonas wittichii RW1	-	535	4100

Continued on next page

Table 3.1 : continued from previous page

Organism name	Gram-positive (+) / Gram-negative (-)	Essential genes in dataset	Total genes in dataset
Staphylococcus aureus N315	+	302	2329
Staphylococcus aureus NCTC 8325	+	351	2759
Streptococcus agalactiae A909	+	317	907
Streptococcus pneumoniae	+	244	127
Streptococcus sanguinis	+	218	2270
Synechococcus elongatus PCC 7942	-	682	2165
Rhodopseudomonas palustris CGA009	-	552	4799
Vibrio cholerae N16961	-	779	3497

3.4 Online gene essentiality database

First introduced in 2012 [32] the Online Gene Essentiality database (OGEE) now contains the data for over 100 large-scale gene essentiality experiments. Including the essential and non-essential genes identified from 65 studies for 39 bacterial organisms [33]. OGEE contains experimentally tested essential and non-essential genes, along with associated gene features such as expression profiles, duplications and conservation across species. This data is combined with text-mining results to produce a list of features for each gene with the genes organised according to their sources [32].

For this study 33 bacterial species were selected, information on the organisms is shown in Table 3.2. For the species selected, the genes obtained from OGEE were matched to a genbank file using the method described in Section 3.2. Of the 33 bacterial species studies 26 are gram negative and 7 gram positive organisms.

As with DEG, we excluded some of the data. Those studies that had different experimental conditions and those that did not include either the essential or non-essential genes. This was because both classes are needed for our evaluation method (described in Section 4.4.1).

Table 3.2: **Information on the 33 bacterial organisms selected from OGEE.** Shown for each organism: whether it is gram-negative (-) or gram-positive (+), the number of essential genes in OGEE and the number of essential and non-essential genes in the final dataset after being matched to Genbank files.

Organism name	**Gram-positive (+) / Gram-negative (-)**	**Essential genes in dataset**	**Total genes in dataset**
Acinetobacter baylyi ADP1	-	293	3195
Agrobacterium tumefaciens str. C58	-	361	5154
Bacillus subtilis subsp. subtilis str. 168	+	228	4160
Brevundimonas subvibrioides ATCC 15264	-	411	3050
Burkholderia cenocepacia J2315	-	162	3376
Caulobacter crescentus NA1000	-	480	3614
Escherichia coli K12 (1)	-	298	4054
Escherichia coli K12 (2)	-	609	3553
Francisella tularensis subsp. novicida U112	-	385	1685
Haemophilus influenzae Rd KW20 (1)	-	450	953
Haemophilus influenzae Rd KW20 (2)	-	491	1364
Haemophilus influenzae Rd KW20	-	14	29
Helicobacter pylori 26695 (1)	-	33	45
Helicobacter pylori 26695	-	310	1362
Mycobacterium tuberculosis H37Rv (1)	+	611	3073

Continued on next page

Table 3.2 : continued from previous page

Organism name	Gram-positive (+) / Gram-negative (-)	Essential genes in dataset	Total genes in dataset
Mycobacterium tuberculosis H37Rv (2)	+	423	3394
Mycoplasma genitalium G37	+	378	470
Mycoplasma pneumoniae M129 (1)	-	362	599
Mycoplasma pneumoniae M129	-	266	496
Mycoplasma pulmonis UAB CTIP	+	404	673
Neisseria gonorrhoeae MS11	-	665	1714
Pseudomonas aeruginosa PAO1	-	336	5515
Pseudomonas aeruginosa UCBPP-PA14 (1)	-	431	5725
Pseudomonas aeruginosa UCBPP-PA14	-	331	4664
Rhizobium leguminosarum bv. viciae 3841	-	280	4230
Salmonella enterica subsp. enterica serovar Typhi str. CT18	-	424	4084
Salmonella enterica subsp. enterica serovar Typhi Ty2	-	2264	4322
Shewanella oneidensis MR-1	-	323	2185
Staphylococcus aureus subsp. aureus NCTC 8325	+	350	2891
Streptococcus pneumoniae D39	+	108	274
Synechococcus elongatus PCC 7942	-	677	2364
Vibrio cholerae O1 str. C6706	-	448	3546

Continued on next page

Table 3.2 : continued from previous page

Organism name	Gram-positive (+) / Gram-negative (-)	Essential genes in dataset	Total genes in dataset
Yersinia pestis KIM	-	606	3481

3.5 Chapter summary

Table 3.3 below shows an overall summary of the two datasets collected. Shown are the number of organisms, instances for each class (essential and non-essential genes) and the gram-positive and -negative organisms. These are all important factors that can affect the prediction accuracy of the machine learning models. The dataset file at this stage contains the nucleotide gene sequences, the gene or protein identifier match in the genbank file and an essentiality class label.

Table 3.3: **Summary of the collected datasets.** Included are the number of: organisms, gram-positive and -negative organisms, essential and non-essential genes.

Dataset	Number of organisms	Number of gram-positive	Number of gram-negative	Essential genes	Non-essential genes
DEG	40	11	29	16481	92370
OGEE	33	7	26	14212	75082

Chapter 4

Determining the Classifier Feature Sets

4.1 Introduction

The first part of this chapter starts by describing the features that were generated for each gene in the dataset and then covers the classification and feature selection algorithms that were applied to create feature subsets. The second part of the chapter describes the method used to apply the feature selection algorithms. There is an analysis of the features present in each generated subset and an evaluation of each feature subsets performance with the identified classifiers.

4.2 Initial feature generation

For each gene in the dataset, 62 sequence derived features were collated, information about the features is shown in Table 4.1. Where indicated, features were computed using additional tools and added to the dataset. The features have been widely used in other prediction studies and, as covered in Section 2.4, can be split into four main types: sequence information (sometimes referred to as intrinsic or gene), homology, subcellular localisation and physicochemical.

Table 4.1: **The 62 features extracted for each gene.** Listed is the feature type, abbreviations, a short feature description and any additional programs used for feature extraction.

Feature type	Abbreviations	General feature description	Tool
Gene	GC	G+C content of the gene	
	STRAND	Positive or negative strand	
	L_aa	Number of amino acids	CodonW [94]
	G3s, T3s, C3s, A3s, GC3s	Base composition at silent sites	
	CAI	Codon Adaptation Index	
	CBI	Codon Bias Index	
	Fop	Frequency of Optimal codons	
	A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y	Individual amino acid usage	
	RARE	Use of rare amino acids	
	TMAA	Percentage of amino acids in helices	TMHMM [95]
	TM60	Percentage of amino acids in helices within the first 60 nucleotides	
Homology	P-3, P-5, P-7, P-10, P-15, P-20, P-30	Paralogs of the gene for a range of e-values	BLAST [96]
Subcellular	TMH	Number of transmembrane helices	TMHMM [95]
	TM60H	Transmembrane helices within the first 60 nucleotides	

Continued on next page

Table 4.1 : continued from previous page

Feature type	Abbreviations	General feature description	Tool
	CELL, CYT, CYTM, EXT, OUTM, PERI	Predicted subcellular localization to the sites: Cellwall, Cytoplasm, Cytoplasmic Membrane, Extracellular, Outer Membrane and Periplasm	PSORTb [97]
Physiochemical	MW	Molecular weight	Pepstats [98]
	ISO	Isoelectric point	
	TINY, SMALL, ALIP, AROM, NONP, POL, CHAR, BASIC, ACID	Molar percentage of the corresponding amino acids for each class: Tiny, Small, Aliphatic, Aromatic, Non-Polar, Polar, Charged, Basic and Acidic	
	ARO	Frequency of aromatic amino acids	CodonW [94]
	HYD	Hydrophobicity of protein	

4.3 Classifier and feature selection algorithms

As covered in the literature review many different classification algorithms and feature selection methods have been applied to the prediction of essential genes. The algorithms selected for this study have been widely used in other studies as shown in Table 2.1.

4.3.1 Classification algorithms

In this study we are comparing commonly used supervised algorithms, where the class of each gene in the training data must be known. The following algorithms are implemented in the Scikit-learn [99] toolbox v0.20.3 and were used with their default parameters:

- Decision Trees (DT)
- Logistic Regression (LR)
- Naive Bayes (NB)
- Nearest Neighbors (kNN)
- Neural Networks (NN)
- Random Forest (RF)
- Support Vector Machine with linear kernel (LIN)

4.3.2 Feature selection algorithms

For the study we applied a range of feature selection methods from all three groups: filter, wapper and embedded. In total 18 feature selection algorithms were applied:

- Correlation-based Feature Selection (CFS)
- Variance (VAR)
- ReliefF
- Least Absolute Shrinkage and Selection Operator (LASSO)
- Recursive Feature Elimination (RFE) - Combined with the DT, LR, RF and LIN classifiers
- Sequential Forward Selection (SFS) - Combined with the DT, LR, kNN, RF and LIN classifiers

- Sequential Backward Selection (SBS) - Combined with the DT, LR, kNN, RF and LIN classifiers

4.4 Feature selection method

To begin building the pipeline for feature selection, we used a simple training and testing method. The collated database of essential genes (DEG) was split 80 / 20 for training and testing respectively. From there we developed the framework below to investigate the feature subsets selected when a feature selection algorithm is run within a cross validation loop.

An outline of the framework developed for each of the 18 feature selection methods is shown in Figure 4.1. Before classification each gene was assigned a Boolean value regarding its essentiality (0 for essential and 1 for non-essential). The class imbalance between the essential and non-essential genes was left because this imbalance naturally exists within organisms. For each feature selection method 10-fold cross validation was applied in which the whole of the DEG dataset was split into 10 parts, 9 parts are assigned as the training data and 1 part is kept separate and is the testing data. The feature selection method was applied only to the training data, the resulting model and the feature selected were saved.

Each classifier was then in turn trained with the training data and the subset of features selected by the method, then used to predict the testing data. Once all classifiers are run, one fold of the cross validation is complete and new training and testing data is assigned. Stratified k-fold was applied so that each fold has the same class proportions.

4.4.1 Evaluation method

To evaluate our classifier models we applied the commonly used Receiver Operating Characteristic (ROC) curve method. A ROC curve is a probability curve which provides information on how well the model is capable of distinguishing between classes. The area under the ROC curve (AUC) is the value calculated for detailed comparisons. The closer the AUC value is to 1, the better the model is at correct predictions.

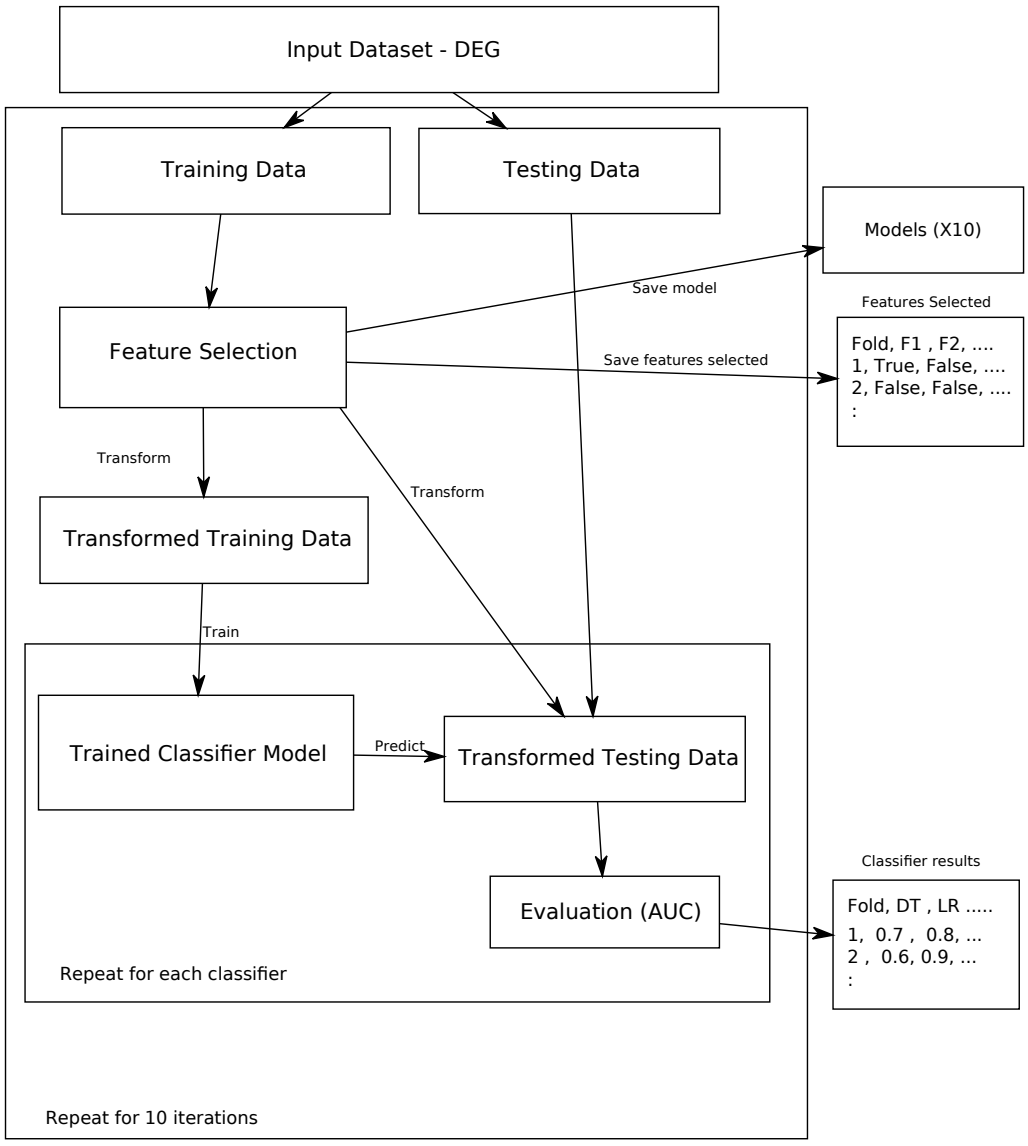

Figure 4.1: **Overview of the developed framework.** Shown is our developed framework in which each feature selection algorithm is applied within a cross validation loop, followed by applying the subset selected to each classifier and evaluation.

4.5 Features subsets generated

For each of the 18 feature selection methods, Figure 4.2 provides information on the resulting subsets. In the central heatmap, each individual cell shows how many times over the whole cross-validation the feature was selected. The maximum for our study being 10. The right frequency plot is a summary of how often each feature was selected over all the feature selection methods. While the bottom plot shows the average number of features, out a maximum of 62, per cross-validation fold for each method.

From the plots we can observe that the top 3 selected features are CAI, C and L_aa. CAI and L_aa have strong proven links to gene essentiality. The least 3 selected features are CYTM, GC3s and Fop.

The top selected features were selected more consistently across the algorithms, i.e. the features were selected for almost every fold of the cross-validation.

RFE-RF selected the most features and VAR the least. Variance selected the same 4 features for every application of the algorithm but 3 of them are not frequently selected by other algorithms. ReliefF also showed little variation in the features selected for each fold. RFE-RF selected all features except RARE at least once.

SFS-KNN and SBS-KNN have almost the same selection pattern.

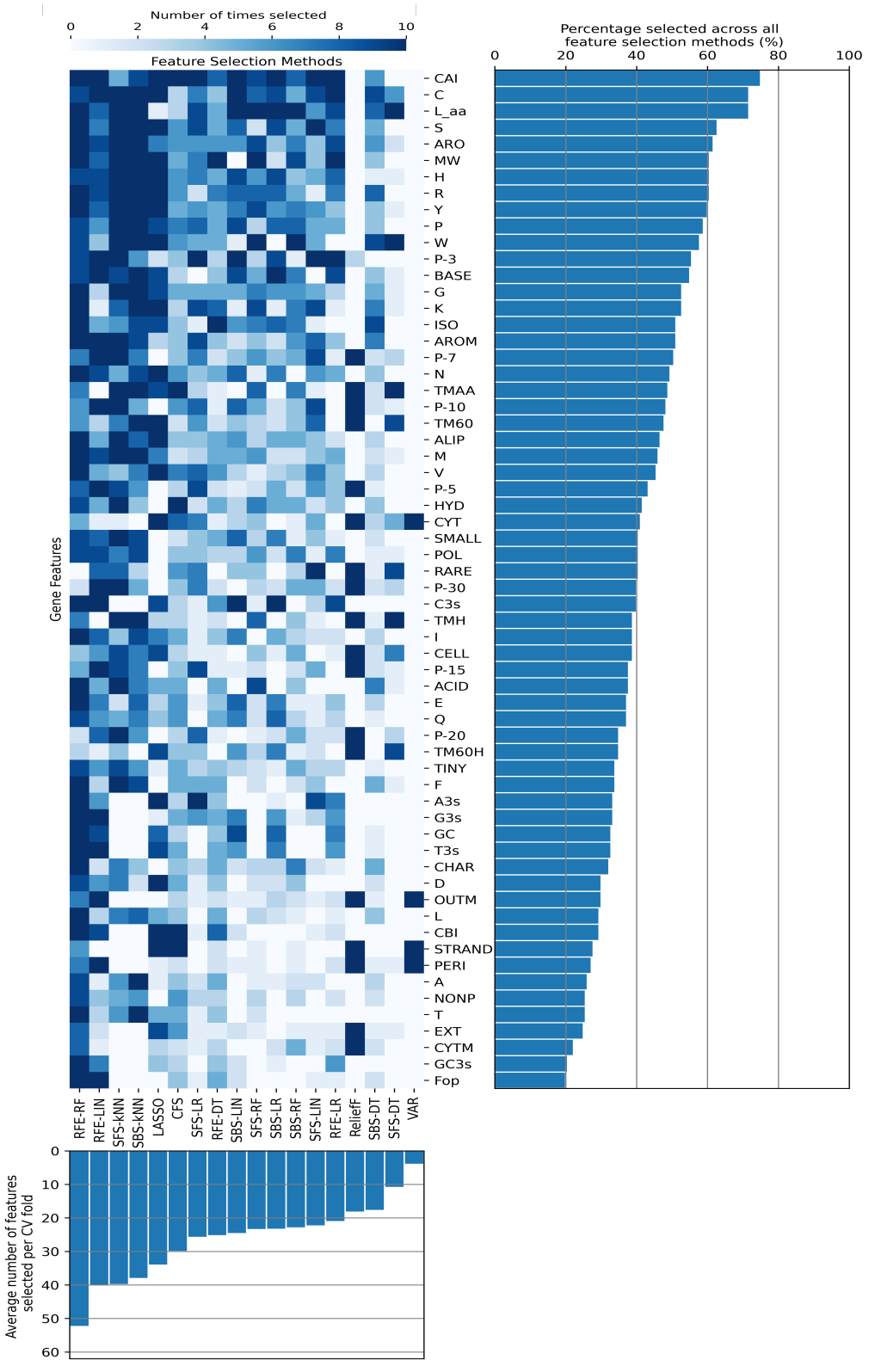

Figure 4.2: **Feature subsets selected by each algorithm.** For each feature selection method the center heatmap shows the frequency at which each feature was selected for the cross validation. The right plot shows the percentage each feature was selected over the whole feature selection method. The bottom plot shows the average number of features selected per cross validation fold for each feature selection algorithm.

4.6 Classifiers with feature subsets results

To study the impact of feature selection on classifier performance during each cross-validation loop we trained each of the seven classifiers in Section 4.3.1 with the selected subset of features and predicted the testing split of data. The framework was also applied to a the whole feature set (all 62 features) to generate a baseline for comparisons.

In total 133 classifier and feature set combinations were tested with 10-fold cross-validation. All of the 1,330 models were evaluated using the ROC curve method described in Section 4.4.1. For each combination the average AUC (avAUC) over the 10 folds and the standard deviations calculated are shown as a heatmap in Figure 4.3.

For the models trained with all features, first column in Figure 4.3, the RF classifier gave the highest avAUC of 0.703, but the other classifiers fall within 1 SD, with the exception of the LIN classifier which is significantly lower.

After applying feature selection the kNN classifier with the SBS-kNN and SFS-kNN subsets produced the highest avAUCs values of 0.722 and 0.715, respectively. Across all the feature subsets RF generally produced the highest avAUC compared to the other classifiers. While the opposite is true for the LIN classifier, which has the lowest results across all 19 feature sets applied, its avAUC fluctuates around 0.5.

Compared to the baseline of All features, the ReliefF and VAR feature subsets caused the greatest reduction to the prediction performance for all seven classifiers. With the exception of these 2 feature subsets, looking at each classifier individually, we can see that there is no feature subset that performs better than another. They all sit within 1 SD of the highest avAUC. The results also show that none of the feature subsets consistently perform the best across all the classifiers.

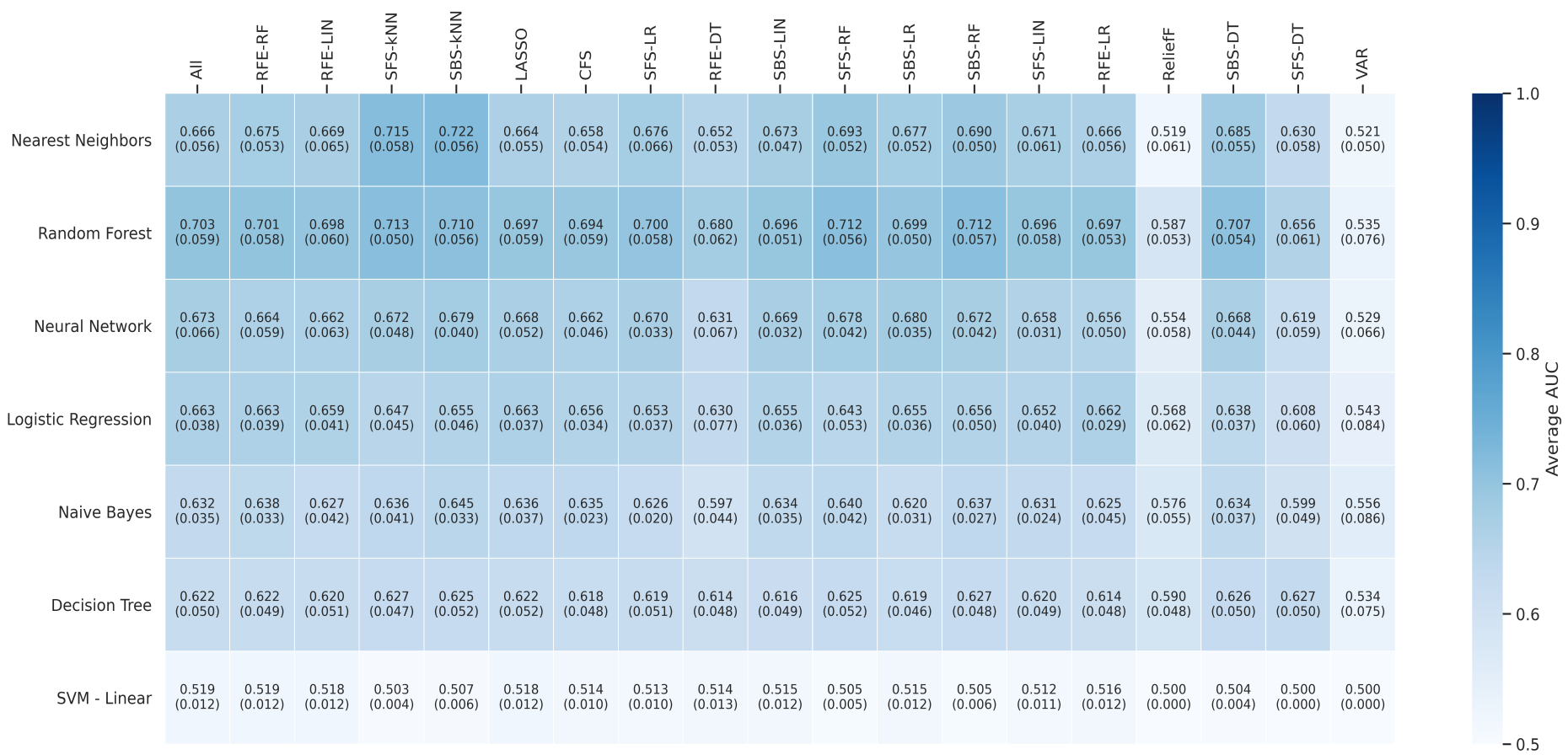

Figure 4.3: **Classifiers with feature subsets AUC results.** A heatmap showing the mean AUC and the standard deviation for each classifier and feature subset combination. Using the DEG dataset and 10-fold cross validation.

4.7 Combining top performing feature subsets

From Figure 4.3 we can observe that for a classifier, there is no single feature subset that performs better than another. For each classifier, all the feature subsets were compared using an independent t-test from the SciPy library [100], with a p-value of 0.05, against the highest avAUC. The results are shown in Table 4.2.

The kNN and LIN classifiers have more distinct results for the feature subsets combinations, as they only have 10 out of 18 that performed best. Whereas for the DT classifier 17 feature subsets performed the best. For RF, NN and LR the feature subset with the highest avAUCs were all different but for all three classifiers the same subsets performed best.

As observed in the heatmap the VAR subset performed significantly worse for all classifiers. With the exception of the DT classifier, the same is true for both the ReliefF and SFS-DT feature subsets.

Table 4.2: **Top performing feature selection subsets for each classifier.** For the each classifier the feature subset producing the highest mean AUC is indicated by a dot (•). Marked by an asterisk (∗) are the feature subsets which are within the standard deviation of the highest mean AUC.

Feature Selection Subsets	**Classifiers**						
	k-Nearest Neighbours	**Random Forest**	**Neural Network**	**Logistic Regression**	**Naive Bayes**	**Decision Tree**	**SVM – Linear**
RFE & RF	∗	∗	∗	•	∗	∗	•
RFE & LIN	∗	∗	∗	∗	∗	∗	∗
SFS & KNN	∗	•	∗	∗	∗	∗	
SBS & KNN	•	∗	∗	∗	•	∗	
LASSO		∗	∗	•	∗	∗	∗
CFS		∗	∗	∗	∗	∗	∗
SFS & LR	∗	∗	∗	∗	∗	∗	∗
RFE & DT		∗	∗	∗		∗	∗
SBS & LIN		∗	∗	∗	∗	∗	∗
SFS & RF	∗	∗	∗	∗	∗	∗	
SBS & LR	∗	∗	•	∗	∗	∗	∗
SBS & RF	∗	∗	∗	∗	∗	•	
SFS & LIN	∗	∗	∗	∗	∗	∗	∗
RFE & LR		∗	∗	∗	∗	∗	∗
ReliefF						∗	
SBS & DT	∗	∗	∗	∗	∗	∗	
SFS & DT						•	
VAR							
Total	10	15	15	15	14	17	10

For each classifier, all of the best performing subsets shown in Table 4.2 were combined to create a single feature set. Figure 4.4 shows the resulting feature sets and how frequently each feature was selected within it.

For all the classifiers every one of the 62 features is present in the combined feature set, this is due to the large number of feature subsets combined. Overall the frequency at which each feature is present largely follows the same pattern as that in Figure 4.2. However, a notable exception to this is the CYT feature for which the frequency decreases. This occurs because it is only highly selected by the worst performing subsets, which were excluded from all of the combined feature sets. Indicating that this feature does not contribute positively to classifier performance.

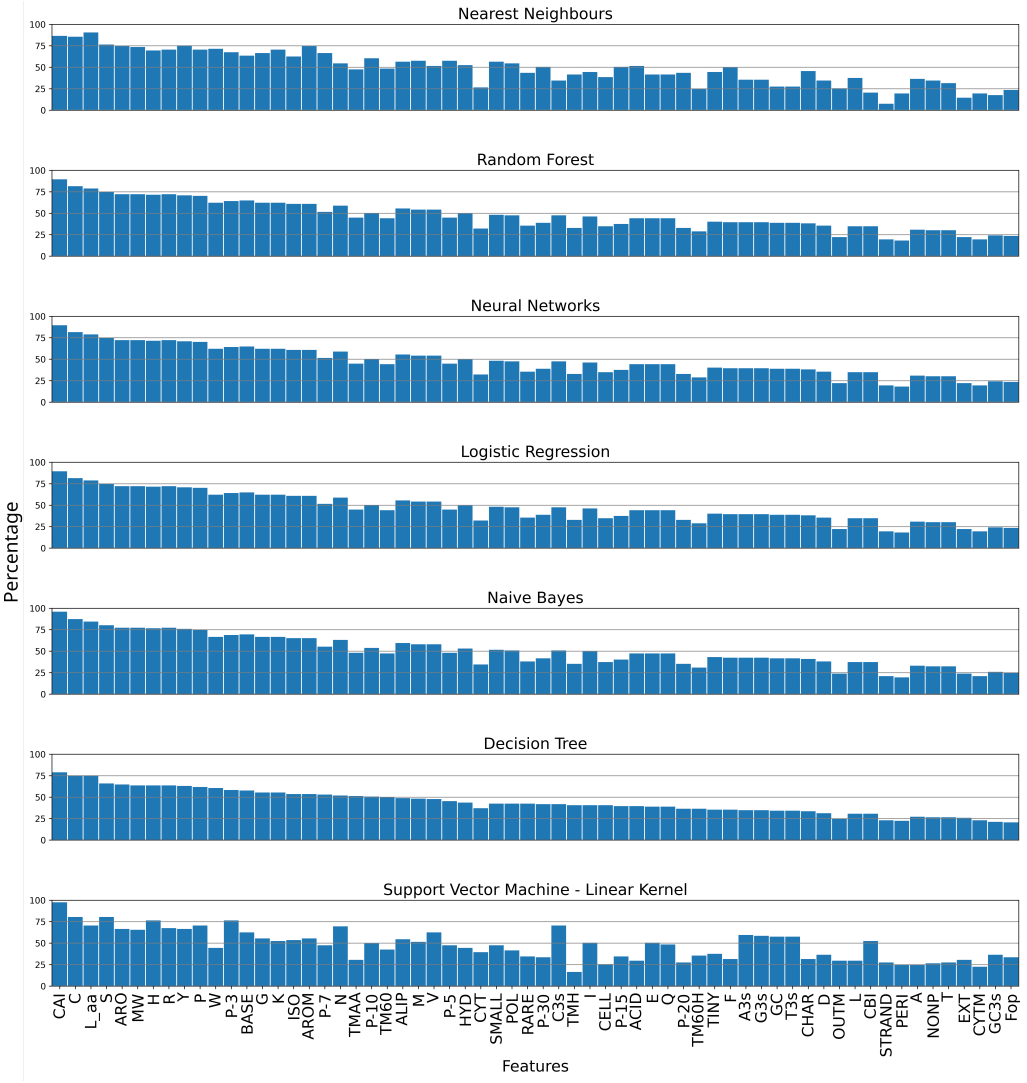

Figure 4.4: **Feature frequency for combined top performing feature subsets.** Shown for each classifier is the frequency at which feature was selected for the combined top performing feature subsets.

4.8 Discussion of feature subsets

In this chapter we investigated different subsets of features and their performance on seven commonly used machine learning classifiers. This was done to provide the best performing classifier and feature subsets for the development of an ensemble model.

To achieve this the first part of the experiment focused on removing redundant features. The removal of redundant features not only aids in decreasing overfitting, and thereby improving the prediction performance of a model, but also reduces the computational complexity speeding up the model. It is important to remember that a feature shown to be linked with gene essentiality may be considered redundant in the presence of another more relevant feature with which it is correlated [101]. This is shown in Figure 4.2 where the feature CAI is the most frequently selected and Fop the least selected. These are both measures of codon optimisation [102], and therefore have the same correlation but Fop becomes redundant. This type of redundancy also applies to the third least selected feature CYTM, protein localization to the cytoplasmic membrane, which is correlated with the transmembrane features TMAA, TMH, TM60H and TM60.

Another highly selected feature, at 72% is the amino acid C, the cysteine amino acid is required for protein stability, which is especially important for proteins exported outside the cell [94]. This allows it to be used as rudimentary feature for protein localization, resulting in EXT only being selected 25%. This has its benefits because calculating C is faster than EXT. It is also possible that C is so frequently selected because it partially replaces the feature TINY.

As expected, the feature L_aa which measures the length of the amino acids sequence is frequently selected at 72% due to its strong correlation with gene essentiality. The essential genes usually have either short or long sequence lengths [63, 64].

GC3s on the other hand, while having strong links to essentiality, is only selected 20% of the the time. This is also seen in a study performed by Liu *et al.* using multiple organisms [80]. The reason for this is because the preference for the 3rd codon is very organism specific [103], for example Saccharomyces prefers AT-ending codons instead of GC [104], therefore this feature is not useful when using multiple organisms for training.

Overall, there were no features that were fully ignored by the feature selection methods. This was expected as all the features used in the study were linked to gene essentiality but also because feature selection was ap-

plied within the cross validation loop and therefore on 10 different datasets. Interestingly, of the 62 features less than a third were selected more than 50% across all the feature selection methods. Showing there is a large amount of redundancy.

While the approach we chose showed us that for some features, which share biological and or functional similarity, the models selected one of these gene features more frequently than the other, we did not look into feature relationships before carrying out feature selection.

As feature relationships can affect some classifiers it is possible that they will have impacted some feature selection methods. Correlated features can affect classifiers in different ways. For example with recursive feature elimination (RFE) and logistic regression, if two correlated features are both present their importance to the model would be low. But if one feature is removed the importance score of the other would need to increase. This would require feature importance to be recalculated after each removal step.

For RFE with random forest or other tree-based models if the correlated features are both useful for prediction, which one is selected is essentially random choice. In this case the feature selection method might contain highly correlated redundant features. While this may not affect the models prediction accuracy it also does not allow us to gain any information about feature importance. Future work into investigating multicollinearity in our feature set may allow us to gain an insight into which redundant features can be excluded before feature selection.

Another interesting line of further investigation from this point would be to look into features which frequently appear together across the different feature selection methods and their impact on the models created.

The second part of the experiment concentrated on the effects of feature selection on classifier performance and identifying the best performing feature subset. From the heatmap in Figure 4.3 we observe that, while there are some classifier and feature subsets which show marginal improvements to the avAUC, there is not an individual one we can say works the best. Even though feature selection does not improve the avAUC, it can maintain the avAUC for a classifier, reducing the computational complexity of a model. We were able to reduce features from 62 to an average of 18 features per cross validation fold.

Support vector machines are useful tools in machine learning and have been widely used in gene prediction studies. However, our results show that the SVM with linear kernel models have no discriminatory ability, i.e. it can not predict whether a gene is essential or not [105]. This is likely because

our data is not linearly separable [106, 107].

Wrapper methods have the advantage of interacting with the classifier during feature selection and can therefore choose more meaningful features [108]. This can be seen in our study for the kNN classifier with the SBS-kNN and SFS-kNN feature subsets which produced the two highest avAUCs of 0.722 and 0.715.

4.9 Chapter summary

In this chapter we generated the initial features and then applied various feature selection algorithms. We then investigated the relationships between the generated feature subsets and seven commonly used machine learning classifiers. We found that there is no single subset of features that performs best across all the classifiers. Based on this outcome we decided to use the combined feature set results from this chapter to develop an ensemble approach to improve the AUC performance.

Chapter 5

Ensemble Method

5.1 Introduction

This chapter describes how the ensemble method was built, trained, tested and validated for essential gene prediction. An ensemble algorithm can be used to improve generalizability over the use of a single estimator. For our ensemble we concentrated on an averaging method, where you build several estimators independently and then average their predictions.

5.2 Ensemble design

Here we describe how the ensemble was designed and built to assess how well the method would perform for cross-validation on the DEG dataset.

5.2.1 Classifier and feature combinations

Carrying forward the results for the combined feature sets, each of the base classifiers in the ensemble were trained using all 62 features.

As well as the combined feature sets we investigated the performance of the most frequently selected features for each classifier. The subsets were created by introducing thresholds of 75%, 50% and 25% to the combined frequency of selection data shown in Figure 4.4. The resulting feature subsets generated at each threshold are shown in Table 5.1. For the ensemble section of the project the support vector machine with linear kernel classifier was excluded. This is because in our initial tests the classifier did not train within a reasonable time frame for our project.

As expected, the threshold subsets for RF, NN and LR are identical as the same feature subsets were combined for these 3 classifiers, and although the NB classifier contained one less subset it has the same subsets across

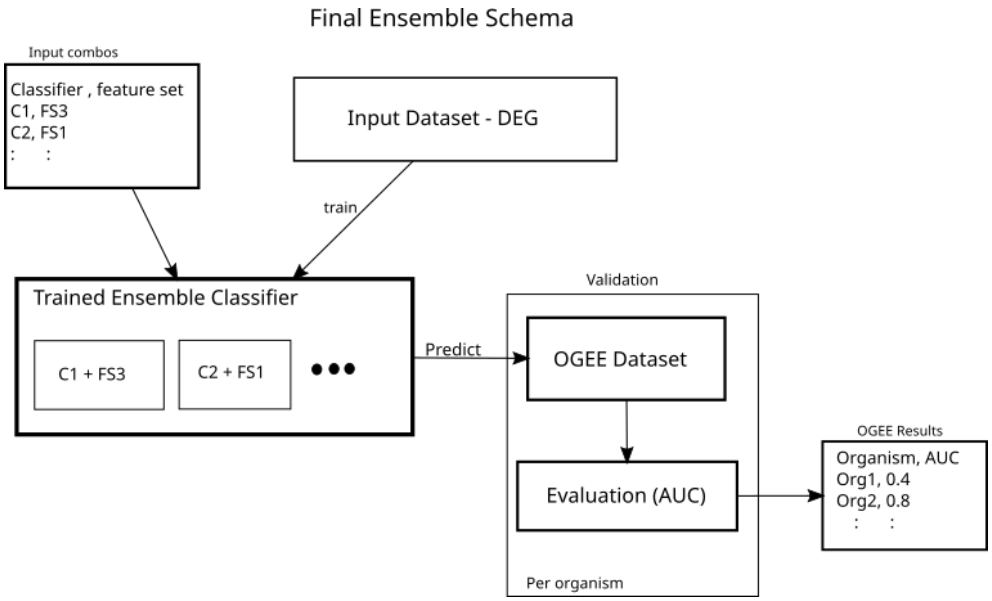

Figure 5.1: Outline of the method used to test the ensemble algorithm on the DEG dataset using 10-fold cross validation.

all the thresholds. The kNN classifier with a threshold of 50% showed the greatest variation for the feature sets and was also the classifier with the most significantly different performances for the feature subsets.

5.2.2 Method

An outline of the method used to test the ensemble algorithm on DEG is shown in Figure 5.1. For this method 10-fold cross validation was applied in which the whole of the DEG dataset was split into 10 parts, 9 parts were assigned as the training data and 1 part was kept separate as the testing data. Each of the base classifiers was trained using the testing data and the relevant feature subset. The features were scaled before training using a Min-Max scaler with a range of -1 to 1. The classifiers had the same default parameters as the feature selection step (the trained classifiers models are also stored). Each trained base classifier was then passed to the ensemble meta-classifier, where they were then used to predict the gene essentialities for the testing data. The performance of the ensemble algorithm within each loop was evaluated using a ROC curve and the AUC. Once all folds are completed the mean AUC and standard deviation (SD) was calculated.

We used a voting classifier ensemble builder from mlxtend [109] with majority voting (hard voting) and each base classifier had equal weighting. Where possible we prefit the base classifiers as the mlxtend algorithm we used does not allow classifiers to be fit in parallel and thus is time consuming.

Table 5.1: Feature sets for each classifier generated by appying thresholds.

Features	75% threshold						50% threshold						25% threshold					
	KNN	RF	NN	LR	NB	DT	KNN	RF	NN	LR	NB	DT	KNN	RF	NN	LR	NB	DT
CAI	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
C	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
L_aa	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
S	*	*	*	*	*		*	*	*	*	*	*	*	*	*	*	*	*
ARO							*	*	*	*	*	*	*	*	*	*	*	*
MW							*	*	*	*	*	*	*	*	*	*	*	*
H							*	*	*	*	*	*	*	*	*	*	*	*
R							*	*	*	*	*	*	*	*	*	*	*	*
Y	*						*	*	*	*	*	*	*	*	*	*	*	*
P							*	*	*	*	*	*	*	*	*	*	*	*
W							*	*	*	*	*	*	*	*	*	*	*	*
P-3							*	*	*	*	*	*	*	*	*	*	*	*
BASE							*	*	*	*	*	*	*	*	*	*	*	*
G							*	*	*	*	*	*	*	*	*	*	*	*
K							*	*	*	*	*	*	*	*	*	*	*	*
ISO							*	*	*	*	*	*	*	*	*	*	*	*
AROM							*	*	*	*	*	*	*	*	*	*	*	*
P-7							*	*	*	*	*	*	*	*	*	*	*	*
N							*	*	*	*	*	*	*	*	*	*	*	*
TMAA												*	*	*	*	*	*	*
P-10							*	*	*	*	*	*	*	*	*	*	*	*
TM60												*	*	*	*	*	*	*
ALIP							*	*	*	*	*		*	*	*	*	*	*
M							*	*	*	*	*		*	*	*	*	*	*
V							*	*	*	*	*		*	*	*	*	*	*
P-5							*						*	*	*	*	*	*
HYD							*						*	*	*	*	*	*
CYT													*	*	*	*	*	*
SMALL							*						*	*	*	*	*	*
POL							*						*	*	*	*	*	*
RARE													*	*	*	*	*	*
P-30							*						*	*	*	*	*	*
C3s													*	*	*	*	*	*
TMH													*	*	*	*	*	*
I													*	*	*	*	*	*
CELL													*	*	*	*	*	*
P-15							*						*	*	*	*	*	*
ACID							*						*	*	*	*	*	*
E													*	*	*	*	*	*
Q													*	*	*	*	*	*
P-20													*	*	*	*	*	*
TM60H														*	*	*	*	*
TINY													*	*	*	*	*	*
F													*	*	*	*	*	*
A3s													*	*	*	*	*	*
G3s													*	*	*	*	*	*
GC													*	*	*	*	*	*
T3s													*	*	*	*	*	*
CHAR													*	*	*	*	*	*
D													*	*	*	*	*	*
OUTM													*					*
L													*	*	*	*	*	*
CBI														*	*	*	*	*
STRAND																		
PERI																		
A													*	*	*	*	*	*
NONP													*	*	*	*	*	*
T													*	*	*	*	*	*
EXT																		*
CYTM																		
GC3s																		
Fop																		
Total	5	4	4	4	4	3	30	23	23	23	23	22	54	55	55	55	55	57

5.3 Results

In Table 5.1 are the results for the ensemble method tested on DEG with 10-fold cross validation. The feature sets investigated are the combined features and thresholds described in Section 5.2.1. For the largest features sets, combined and 25%, the RF classifier generates the highest avAUC with values of 0.690 and 0.745 respectively. Whereas, for the smaller 75% and 50% sets the highest avAUCs of 0.849 and 0.860 were produced by our ensemble model. The results for the combined, 75% and 25% feature subsets all sit within 1 SD of the highest avAUC.

The resulting highest avAUC combination was the ensemble model with the 50% feature subset with a value of 0.860, with the three other ensemble models within 1 SD. For the 50% feature subset the NN, LR and NB classifiers performed worse, i.e. their avAUCs sit outside 1 SD, than our ensemble model.

Table 5.2: Performance of the ensemble method on the DEG dataset using the subsets of combined features, and the thresholds created by applying the thresholds 75%, 50% and 25%. The average AUC (avAUC) for all 10-folds and the standard deviation (SD) are shown. In bold are the highest average AUC scores for each subset.

Classifiers	**Combined**		**75%**		**50%**		**25%**	
	avAUC	SD	avAUC	SD	avAUC	SD	avAUC	SD
NN	0.556	0.137	0.602	0.072	0.675	0.051	0.562	0.137
KNN	0.572	0.104	0.747	0.057	0.798	0.070	0.605	0.109
LR	0.626	0.070	0.622	0.093	0.653	0.057	0.642	0.073
RF	**0.690**	0.118	0.843	0.096	0.858	0.101	**0.745**	0.113
NB	0.618	0.070	0.611	0.090	0.650	0.053	0.627	0.065
DT	0.601	0.081	0.804	0.078	0.808	0.076	0.605	0.079
Ensemble	0.665	0.129	**0.849**	0.092	**0.860**	0.085	0.690	0.131

5.4 Discussion of ensemble DEG results

As mentioned in Section 2.4.1, through combining multiple classifiers it is possible to achieve more accurate predictions compared to using single classifiers. It is based on the idea that an error in a single inducer will be compensated for by the other classifiers and improve the prediction performance [72, 73]. This can be clearly observed in the results for the models built using the 75% and 50% threshold feature sets, Table 5.2, where the ensemble avAUCs are higher than those for the individual base learners.

Through combining the top performing subsets for each classifier and studying the most selected features we were able to improve the AUC performances for all 6 base learners. We were able to achieve this performance increase while simultaneously reducing the number of features from 62 to 22 - 30, Table 5.1. This idea of improving classifier performance with reduced features is the focus of feature selection.

The RF classifier can perform better, than other classifiers, with a larger number of features as it carries out feature selection and ranks the features as part of its classification and it is also an ensemble. It is also less affected by multicollinearity. This may explain why is performs better, than the other base learners, with the higher number of features in the combined and 25% threshold feature sets.

The number of features for the base classifiers is between 22 - 30, interestingly for the reliefF and SFS-DT subsets which selected around the same number of features at 19 and 26, the performance of the subsets was significantly worse than all the others. This highlights the importance of the features used to train the classifiers, even if they have links to essentiality.

A comparison of the ensemble models performance with previous studies is shown in Table 5.3. Overall, for 40 bacterial studies our ensemble method was able to reduce 62 to 23 - 30 and improve the avAUC to 0.860. In comparison, Liu et al. [80] used LASSO to reduce the number of features from 57 to 40, while maintaining the AUC of all features. Using an optimised SVM trainied with 31 bacterial organisms they achieved an AUC of 0.794.

Deng et al., applied an ensemble where all 4 base classifiers were trained with the same feature sets. They were able to achieve AUCs of 0.89 and 0.93 using only 10 and 13 feature respectively. However their method was applied using cross validation within the same organisms. Therefore the datasets were small compared to ours and previous studies, with only 3308 and 4289 instances available for training and testing.

From Table 5.3 we observe that our model produces higher AUCs for

Table 5.3: Performance comparison of ensemble model with previous studies.

Study	**Classifiers**	**Number of features**	**Number of organisms**	**AUC**
Our model	Ensemble (Naive Bayes, kNN, Decision Tree, Logistic Regression, Neural Network & Random Forest)	23-30	40	0.860
Liu et al., 2017 [80]	SVM	40	31	0.794
Ning at al., 2014 [56]	SVM	158	16	0.76
Xu at al., 2020 [110]	Neural Network	29	31	0.773
Saha et al., 2006 [51]	Ensemble (Weighted kNN & SVM)	13	1	0.82
Deng et al., 2011 [40]	Ensemble (Naive Bayes, Logistic Regression, Decision Tree, & CN2 Rule)	13	1	0.93
		10	1	0.89

previous studies also utilising large datasets for generalizability [56, 80, 110]. The AUC for our ensemble model matches previous studies that apply ensemble methods [40, 51].

5.5 Ensemble validation

This is a test to see how well the ensemble models generalise to the organisms from the online gene essentiality database (OGEE) dataset.

5.5.1 Method

In the ensemble framework each of the base classifiers was trained using the whole of the DEG dataset and the relevant feature subset. The features were scaled before training using a Min-Max scaler with a range of -1 to 1. The classifiers had the same default parameters as the feature selection step (the trained classifiers models were also stored). Each trained base classifier was then passed to the ensemble meta-classifier, implemented from mlxtend, where they are then used to predict the gene essentiality for the individual organisms within the OGEE dataset. The base classifiers predict the probabilities of each gene being essential or non-essential. The ensemble model combines the results from each base classifier and then votes to produce a essentiality prediction. Our ensemble model uses majority voting (hard voting) and each base classifier has equal weighting. The performance of the ensemble model is evaluated using the AUC value for a ROC curve. The confusion matrix for each classifier was saved so other evaluation methods can be calculated, e.g accuracy or specificity. Data to generate the ROC curves was also stored.

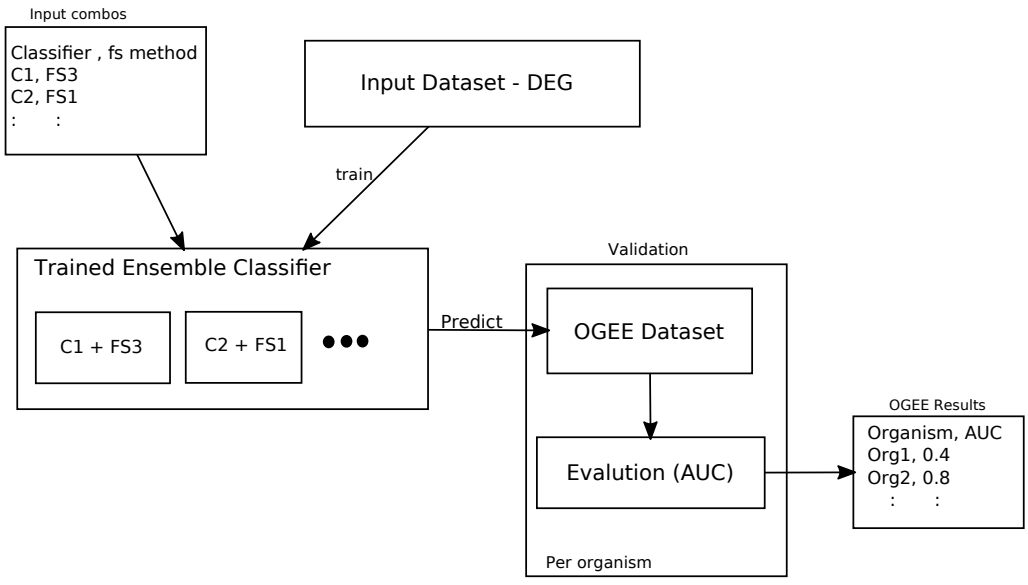

Figure 5.2: Outline of the method used to test the performance of the ensemble algorithms on the OGEE dataset organisms.

5.5.2 Results

Every organism in the OGEE dataset was predicted with each base classifier and subset model and the AUC scores are shown in the sections below. As both of the online databases contain data on model organisms, 25 out of 33 of the OGEE organisms are in the DEG dataset (using the names provided to match). To aid understanding of the results the remaining organisms in OGEE have been split into the following three groups:

- Same species - most closely matched to an organism in the DEG training set, *Salmonella enterica* CT18
- Same genus - *Burkholderia cenocepacia*, *Francisella tularensis* and *Mycoplasma pneumoniae*
- No matches - the least closely matched to any organisms in the training set, the results for these are of the most interest, *Neisseria gonorrhoeae*, *Rhizobium leguminosrum* and *Yersinia pestis*

Combined feature sets

The results for the classifier and combined feature sets are shown in Figure 5.3. As expected from Table 5.2 the RF classifier generally has the highest AUC values, followed by our ensemble model. The NB classifier generally has the worst AUC performance.

For the same species group RF had a higher AUC than the ensemble. The same applies to *B. cenocepacia* and *F. tularensis* from the same genus group. However for *M. pneumoniae* the NN clasifier had the highest AUC. For the group with no matches, for 2 of the organisms NB gives the best performance, but for one it was the ensemble.

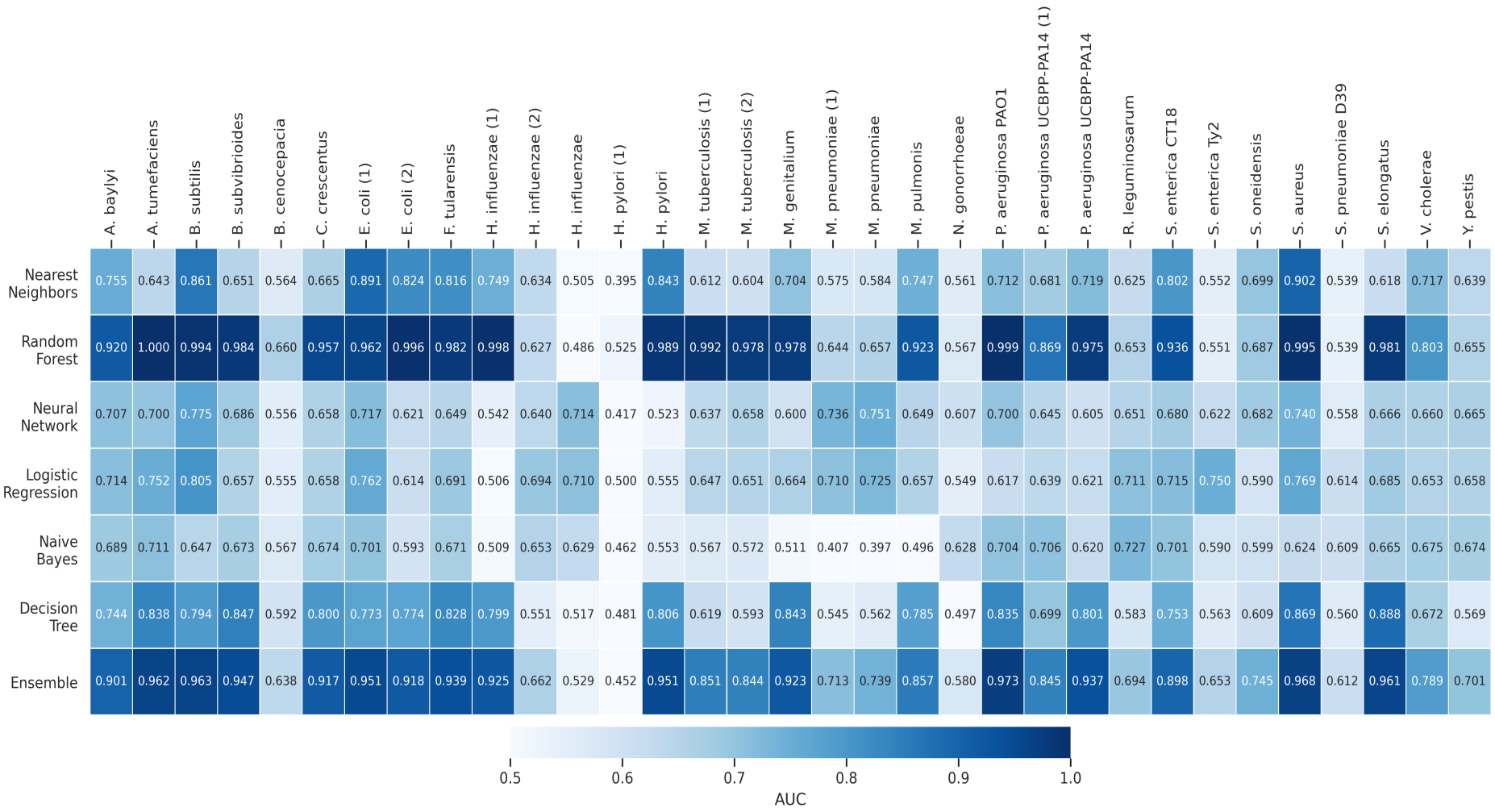

Figure 5.3: The AUC results for each OGEE organism for all features.

Thresholds

The results for the classifier and threshold subsets are shown in Figures 5.4 - 5.6. For the 75% feature subset the general performance of the classifiers match the DEG cross validation results, with the RF, DT and ensemble classifiers performing the best. The prediction performance of the kNN classifier is also improved.

For the 50% feature subset the general performance of the classifiers match that of the 75% feature subset with the RF, DT and ensemble classifiers performing the best. The NN, RF and NB generally have the lowest performances. The AUC values for the organisms are overall higher than the 3 other feature sets. For all of the organisms in the third group which had no matches in DEG, the ensemble model has the best performance, producing AUC scores of 0.645, 0.805 and 0.757. It also has the best predictions for *M. pneumoniae*.

At a threshold of 25%, the all the models produce worse AUCs. As expected the RF classifier has good prediction performance for the organisms. The NN and NB classifiers produce the worst predictions for the validation data. The best predictions for *M. pneumoniae*, are produced via the 75% and 25% NN classifier.

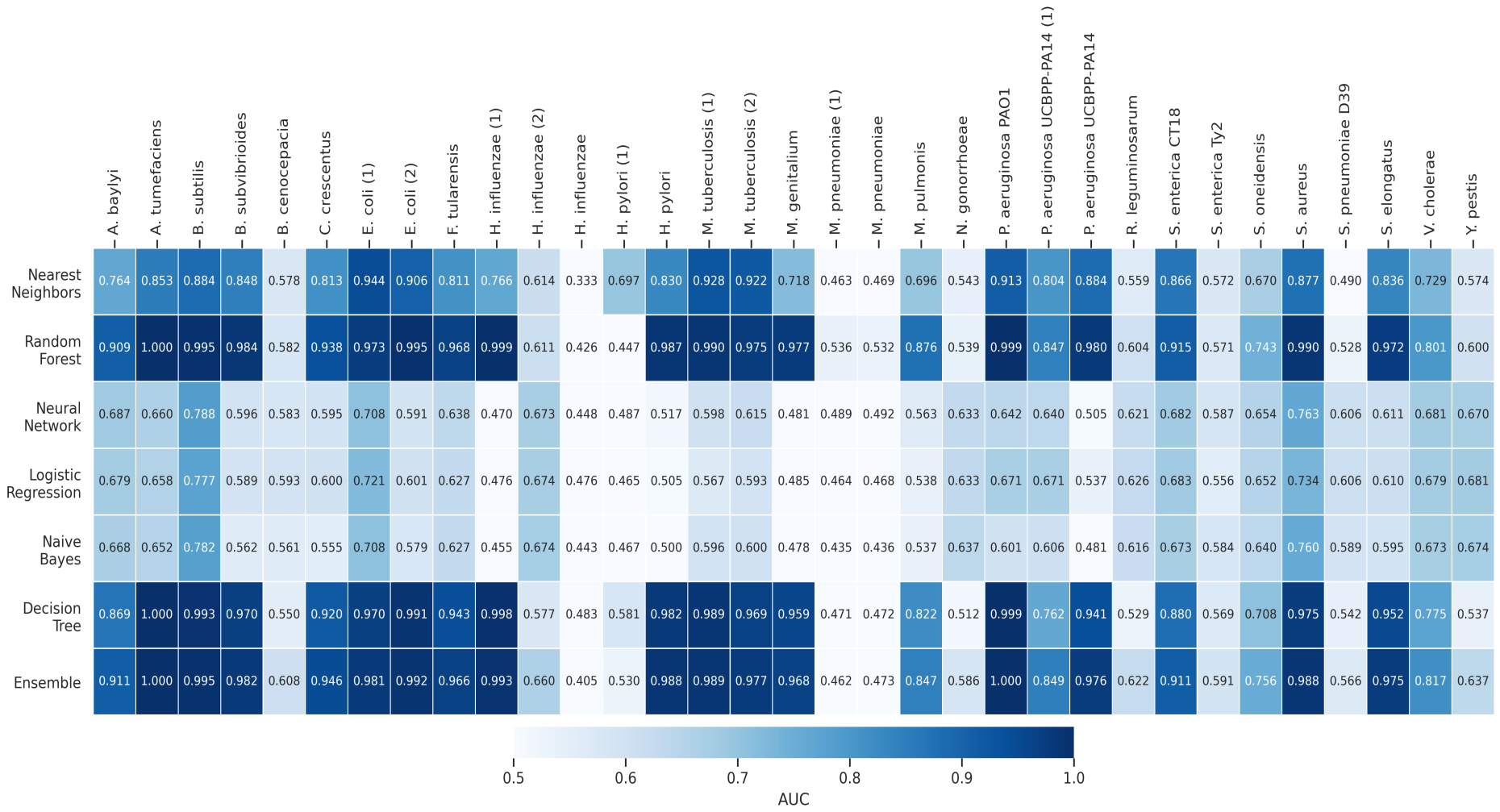

Figure 5.4: The AUC results for each OGEE organism with a 75% threshold.

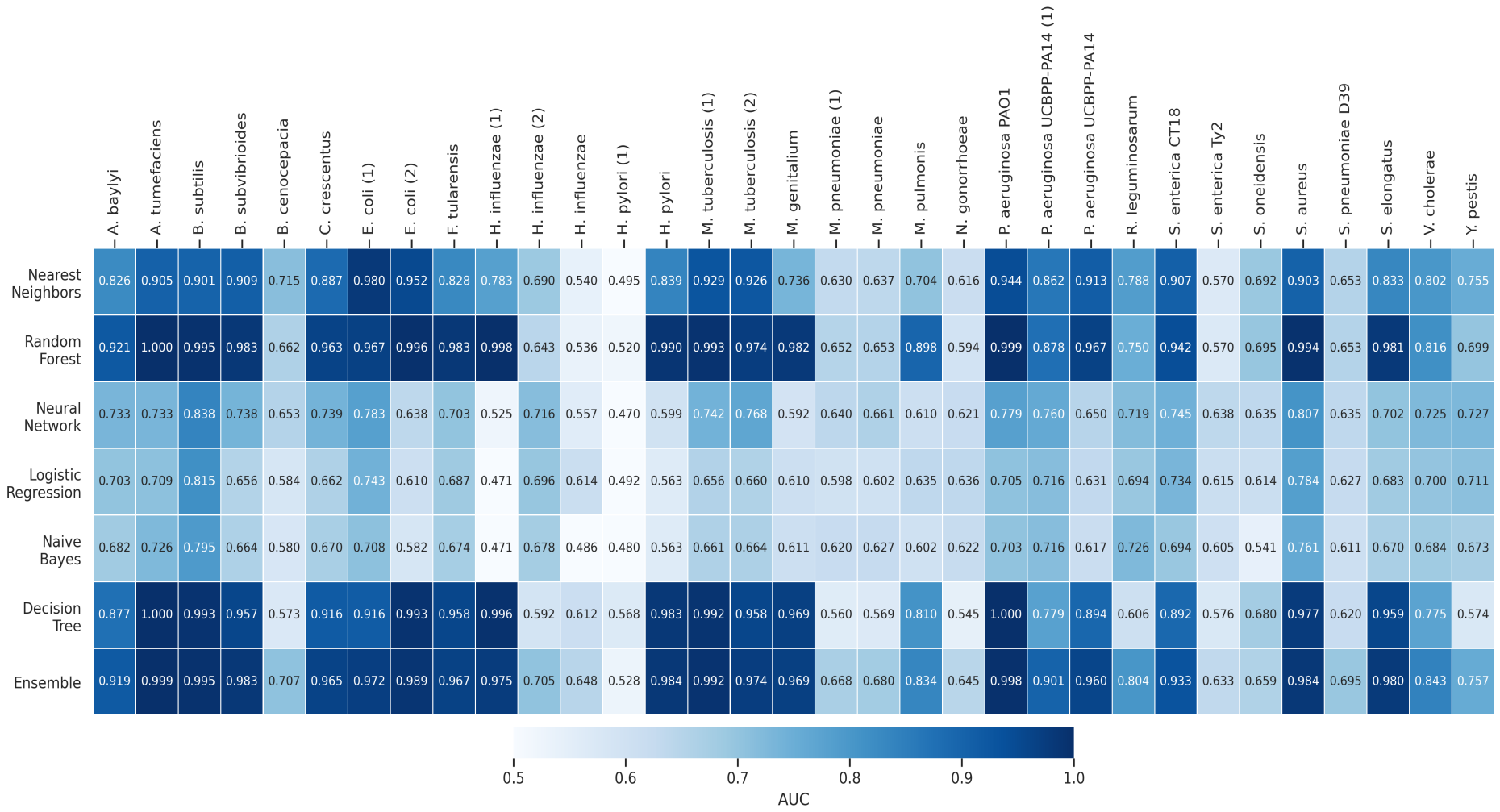

Figure 5.5: The AUC results for each OGEE organism with a 50% threshold.

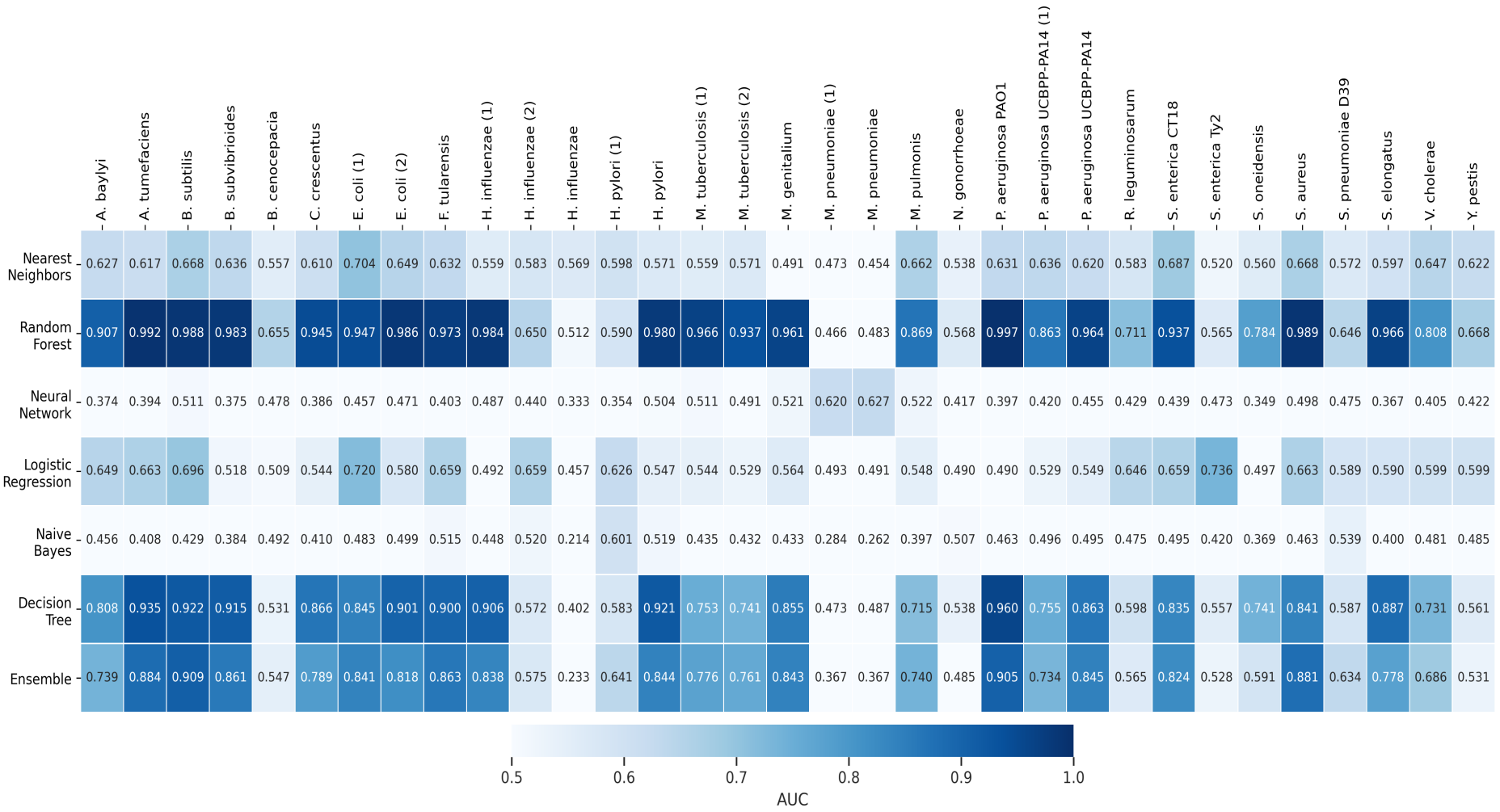

Figure 5.6: The AUC results for each OGEE organism with a 25% threshold.

5.5.3 Discussion of ensemble validation

In this part of the project we set out to validate the ensemble models created with previously unseen data, the OGEE dataset. We were unable to validate our models by running them on the same datasets as previous studies as the data within these studies was available to us within the time-frame of the project. While the papers do contain the organism names, they do not contain the NCBI accession or version IDs which allow to use the exact data. Where DEG has been used previous versions of the database were unavailable to us, as was the version history.

For all 4 ensemble methods the model built with the 50% threshold features produced better results for the validation data overall. This result fits with the general theory of feature reduction improving the generalization ability of a classifier. This point can be further observed in the large AUC increase overall for the kNN and DT base classifiers at the 75% and 50% subsets.

The RF classifier has better predictions for organisms already in the training dataset but not for those that are not. This is most likely because the classifier is over-fitting due to the default parameters in scikit-learn [111].

Looking at the base classifiers, for the organisms not in the training data, we observe that for each organism a different base classifier performs the best. However, when the results are combined in the ensemble we observe that combining classifiers can compensate for the errors in others. Of the 7 organisms, 5 have higher ensemble AUCs than those of the base inducers. The exceptions are *S. enterica* CT18 which is expected as it is a closely related to the strains in the training data, so the RF classifier has the best AUC. Interestingly, the other exception is *B. cenocepacia* for which the kNN classifier performs better. Applying an ensemble method compensates for these differences in classifier preference for specific organisms and still produce reasonable predictions for a wide range of unknowns.

There is an anomalous low AUC spot for the *H. influenzae* and *H. pylori* organisms, this is most likely due to data in the validation set being determined under different conditions for that of the training set. This is one of the main limitations of computational methods where the conditions for the experimental studies must be thoroughly checked.

Looking at the combined set of all features against the 25% threshold we see an interesting pattern where the removal of just a few features causes a large decrease in AUC for the NB and NN classifiers.

A comparison of the ensemble validation results to previous studies is

Table 5.4: Performance comparison of ensemble model with previous studies.

Study	**Classifiers**	**Number of features**	**Number of training organisms**	**AUC (avAUV)**
Our model	Ensemble (Naive Bayes, kNN, Decision Tree, Logistic Regression, Neural Network & Random Forest)	23-30	40	0.645 - 0.933 (0.742)
Liu et al., 2017 [80]	SVM	40	30	0.531 - 0.901 (0.710)
Deng et al., 2011 [40]	Ensemble (Naive Bayes, Logistic Regression, Decision Tree, & CN2 Rule)	9	1	0.69
		10	1	0.80
		-	1	0.80
Xu at al., 2020 [110]	Neural Network	29	30	0.534 - 0.839 (0.698)
Hua et al. [112]	SVM	24	24	0.786
Azhagesan et al,. [113]	Random Forest	100	26	0.566 - 0.911 (0.788)

shown in Table 5.4. Our AUC scores for our validation results, for the 7 organisms not in the training dataset, range from 0.645 - 0.933 with an avAUC 0.742. Liu *et al.* evaluated using a leave-one-genome-out (LOGO) out method and achieved avAUCs of 0.710. An important note for the LOGO method is that the validation data (the genome left out) is present during feature selection.

Deng *et al.* compared 3 organisms one against the other and achieved AUCs of 0.69 - 0.80, using an ensemble method of 4 base classifiers trained with the same feature sets. The method optimises the features selected for each comparison.

The avAUC of our ensemble model sits in the middle of other studies that also use a large number of training organisms to improve generalisation [80, 110, 112, 113].

5.6 Chapter summary

In this chapter we applied thresholds to the combined feature sets to investigate the performances of the most frequently selected features. These subsets were used to create an ensemble method which was built and trained using the DEG dataset and tested with 10-fold cross validation. Then the ensemble model was validated with individual organisms from the OGEE database. Our results showed that for the model built with features selected at a frequency of at least 50% the ensemble model performed better than the base learners. This model also produced the highest AUC scores for the unseen validation data. Our method is able to generalise to unseen data as well as previous studies while requiring less input information than other models. This is an important point for a models application to new organisms for which little to no information is available.

Chapter 6

General Discussion and Conclusions

6.1 Introduction

This final chapter brings together evaluations and discussions from the previous chapters for a general discussion of the thesis. We outline some future directions for research based on this project and some final conclusions.

6.2 General discussion

The motivation for this project was to provide a faster way to identify gene essentiality in organisms of interest in synthetic biology. Our approach to this, and the core aim of this thesis, was to develop an ensemble method using multiple classifiers and feature sets. To achieve this a number of objectives needed to be achieved and this section discusses the progress made towards these objectives.

In Chapter 2 we addressed the first objective and identified classifiers and features commonly applied to the computational identification of essential genes by conducting a thorough literature search. From this search we collated a list of 62 sequence based and derived features and 7 commonly used classifiers. One of the problems identified from the literature was the use of small datasets in the studies. Training sets of a small number of organisms, usually one or two, result in models overfitting and not generalizing well to unseen data.

To address this problem and produce a model that could be used to predict essentiality in a range of organism, we needed to gather as much essentiality data as possible. Chapter 3 describes the labeled data we gath-

ered to satisfy the second objective. We were able to gather 40 organism studies from DEG to use for building and training models and a further 33 studies from OGEE to validate the model. We only used the two online databases as it is difficult to match conditions between individual studies as there is no standardized approach for reporting results making paper mining difficult and time consuming. The DEG and OGEE databases primarily contain studies based on the 'ideal' conditions we use for our definition of essential genes.

Chapter 4 was an important step towards achieving our core aim. The first half of the chapter describes the 62 features generated for each gene. In the second half of the chapter we applied 18 different feature selection methods and studied how often each feature was selected. We then went on to study the AUC performance of these generated subsets on 7 classifiers. Our results showed that there was no single feature subset that performed significantly better than another, for the classifier there were between 10 to 17 feature subsets that performed best. While our methods allowed us to see which features were more frequently selected they do not allow us to gain meaningful insight into feature importance.As understanding this aspect of machine learning could help us improve our ensemble models it is an important line of future research.

The best performing subsets for each classifier were combined and carried forward to develop our ensemble method described in Chapter 5. We applied thresholds to the combined feature sets to generate subsets of features that were selected at a frequency of least 75%, 50% and 25%. The 3 threshold feature sets and combined feature set were then used to build and train 4 ensemble models, along with the DEG dataset. Evaluation of the 4 ensemble models through 10-fold cross validation showed that feature selection improved the AUC scores across all 7 classifiers. Our also showed that for the 75% and 50% threshold subsets the ensemble models perform better than the individual base learners. Of all the classifiers and feature set combinations the ensemble model trained with features selected at least 50% of the time produced the highest avAUC of 0.860. Compared to the literature our ensemble model produces higher AUCs than other studies utilising large datasets for generalizability [56, 80, 110]. Also the avAUC for our ensemble model matches previous studies that apply ensemble methods [40, 51].

All 4 ensemble models were validated using individual organisms from the OGEE dataset. It is important to note that because some of the organisms in the OGEE dataset were also present in the DEG training dataset the AUC scores for these organisms is very high. As the results for these

organisms do not provide a true evaluation of the ensemble models our validation results focus on the 7 organisms not present in the training dataset. As expected of all 4 ensemble models the model trained with features selected at least 50% of the time performed best. For the 7 organisms of interest the model produced AUC scores of 0.645 - 0.933, overall our ensemble models performs better than previous studies which applied a single optimised classifier [80,110]. While the AUCs of our model sit in the middle of other studies that use a large number of training organisms to improve generalisation [80, 110, 112, 113] our model requires less input information making it an ideal candidate for the prediction of genes in new or hard to culture bacterial organisms.

6.2.1 Discussion of evaluation matrix

As covered in section 4.4.1, we used the area under the ROC curve to evaluate our classifier models. This is commonly used for evaluating binary classifiers and is calculated using the false positive and true positive rates. A limitation of this metric is that for imbalanced classes the model may seem more useful for predictions that it actually is. This is because when dealing with imbalanced classes the classifier can predict everything as the larger class and still give good performance when measured using the ROC metric. In these cases precision-recall curves can provide a better insight into performance as it is a measure of how good the model is at predicting the positive class, in our case it would be the essential genes as the minority class [114].

Our research was based on previous studies in which the ROC curve metric was applied to measure the performance of models with balanced classes. However during the project it was decided that for laboratory experiments identifying the non-essential genes is equally important for targeting genes and as a result we choose not address the class imbalance. This meant our classes had a ratio of 1:6, essential to non-essential genes. Due to the context of the research changing, applying the Precision-Recall AUC metric would have provided a deeper insight into the performance of our models. As this cannot be addressed with the time and funding available it should form the start point of any future work.

6.3 Future work

6.3.1 Further development of ensemble

Although the aims of our project were addressed, there remains areas of the method which could be improved and also factors that may improve the performance of the ensemble model. We would firstly need to incorporate a support vector machine classifier, with different kernels, to base inducers. SVMs can produce models with good generalisation and are widely used for gene prediction. Further investigations into feature subsets at smaller frequency of selection thresholds could produce insight into which types of features perform best with particular classifiers.

In important aspect is the voting system applied to the base learners. The results can be combined in a number of different ways, including different voting systems and the base classifiers can be given different weights, i.e. the results of some classifiers hold more importance than than others. Weights could be assigned depending on the classifiers overall performance.

6.3.2 Evolutionary distance

The problem of predicting essential genes as the evolutionary distance increases between the training organisms and target organism is a ongoing problem in synthetic biology. An interesting study would be to evaluate how not only the ensemble performs with increasing evolutionary distance but also the base learners. Increasing the voting weights of classifiers depending on evolutionary distance may increase the scope of our model.

6.4 Conclusions

In conclusion, an ensemble method for the prediction of essential genes in bacteria has been successfully constructed and validated. Through the development of the method, we uncovered interesting relationships between features linked with gene essentiality through subsets not previously suggested in the literature. The results in this work will help speed up promising investigations into the production of greener and sustainable petrochemical replacements.

Glossary

Amino Acid Molecules which have the same basic structure but a different R-group for each of the 20 amino acids. Amino acids sequences or chains are determined by the gene sequence.

Codon A sequence of 3 nucleotides which correspond to a specific amino acid during the synthesis of a protein.

Gene A sequence of base organic molecules (see nucleotides) that encode for the synthesis of product.

Genome The complete set of DNA within an organism.

Helix / Helices In this project we specifically discuss protein helices, which are a spiral chain of amino acids within a protein.

mRNA A molecule produced from the original DNA version of a gene from which a protein can be made. Unlike DNA, mRNA is able to leave the protected nucleus of a cell, where all the genetic information is contained, and move into areas where proteins are synthesized.

Nucleotides Organic molecules which are the units of DNA and RNA (see mRNA). There are 4 bases: adenine (A); thymine (T); cytosine (C) and guanine (G). They form base pairs of A-T and G-C.

Paralogs Copies of a gene on the same genome.

Protein Molecules made of one or more chains of amino acids.

Sequencing We use this term to refer to the process of determining the order of nucleotides within DNA. There are many different methods for this process but as they are not relevant for our project, we use it as an umbrella term to encompass the processes in general.

Subcellular localisation Cells are split into distinct regions. Some proteins have functions distinct to a region and are therefore found there in higher numbers.

Transmembrane A term used to describe a molecule which spans the whole membrane. Meaning that an end of the molecule is present on either side of the membrane.

Expression Also referred to as gene expression, expression is a measure of how often a gene is copied from the genome into mRNA.

Chapter 7

Appendix

A - Organisms excluded from DEG dataset

Listed in Table 7.1 are the bacterial studies excluded from the final DEG dataset.

Table 7.1: Organisms excluded from the DEG dataset.

Organism	**Reason for exclusion**
Streptococcus pneumoniae	No non-essential genes present
*Pseudomonas aeruginosa*PAO1	Required in the presence of Tobramycin
Mycobacterium tuberculosis H37Rv II	Required for cholesterol metabolism
Salmonella enterica serovar Typhi	Required for bile acid tolerance
Streptococcus pyogenes MGAS5448	Todd-Hewitt medium
Streptococcus pyogenes NZ131	Todd-Hewitt medium
Salmonella enterica serovar Typhimurium 14028S	Required for bile acid tolerance

Bibliography

[1] Elizabeth Pennisi. Synthetic genome brings new life to bacterium, 2010.

[2] Daniel G Gibson, John I Glass, Carole Lartigue, Vladimir N Noskov, Ray-Yuan Chuang, Mikkel A Algire, Gwynedd A Benders, Michael G Montague, Li Ma, and Monzia M Moodie. Creation of a bacterial cell controlled by a chemically synthesized genome. *science*, 329(5987):52–56, 2010.

[3] Shimyn Slomovic, Keith Pardee, and James J Collins. Synthetic biology devices for in vitro and in vivo diagnostics. *Proceedings of the National Academy of Sciences*, 112(47):14429–14435, 2015.

[4] Shibin Zhou. Synthetic biology: bacteria synchronized for drug delivery. *Nature*, 536(7614):33–34, 2016.

[5] Jan Claesen and Michael A Fischbach. Synthetic microbes as drug delivery systems. *ACS synthetic biology*, 4(4):358–364, 2015.

[6] M Levin, J Bongard, and J E Lunshof. Applications and ethics of computer-designed organisms. *Nature Reviews Molecular Cell Biology*, 2020.

[7] Stefan Schütz, Bernhard Weißbecker, Peter Schroth, and Michael J Schöning. Linkage of Inanimate Structures to Biological Systems—Smart Materials in Biological Micro-and Nanosystems. In *Smart Materials*, pages 149–157. Springer, 2001.

[8] Timothy S Gardner. Synthetic biology: from hype to impact. *Trends in biotechnology*, 31(3):123–125, 2013.

[9] Charles S Cockell. Synthetic geomicrobiology: engineering microbe–mineral interactions for space exploration and settlement. *International Journal of Astrobiology*, 10(4):315–324, 2011.

[10] Cyprien N Verseux, Ivan G Paulino-Lima, Mickael Baqué, Daniela Billi, and Lynn J Rothschild. Synthetic biology for space exploration: promises and societal implications. In *Ambivalences of Creating Life*, pages 73–100. Springer, 2016.

[11] Gregory Stephanopoulos. Challenges in Engineering Microbes for Biofuels Production. *Science*, 315(5813):801 LP – 804, feb 2007.

[12] James E Bailey. Toward a science of metabolic engineering. *Science*, 252(5013):1668–1675, 1991.

[13] Shota Atsumi, Taizo Hanai, and James C Liao. Non-fermentative pathways for synthesis of branched-chain higher alcohols as biofuels. *Nature*, 451(7174):86–89, 2008.

[14] Shota Atsumi, Anthony F Cann, Michael R Connor, Claire R Shen, Kevin M Smith, Mark P Brynildsen, Katherine J Y Chou, Taizo Hanai, and James C Liao. Metabolic engineering of Escherichia coli for 1-butanol production. *Metabolic engineering*, 10(6):305–311, 2008.

[15] Gregory Stephanopoulos. Metabolic engineering: perspective of a chemical engineer. *American Institute of Chemical Engineers. AIChE Journal*, 48(5):920, 2002.

[16] Toru Jojima, Masayuki Inui, and Hideaki Yukawa. Production of isopropanol by metabolically engineered Escherichia coli. *Applied microbiology and biotechnology*, 77(6):1219–1224, 2008.

[17] H A George, J L Johnson, W E C Moore, L V Holdeman, and J S Chen. Acetone, Isopropanol, and Butanol Production by Clostridium beijerinckii (syn. Clostridium butylicum) and Clostridium aurantibutyricum. *Applied and Environmental Microbiology*, 45(3):1160 LP – 1163, mar 1983.

[18] Dan Ferber. Microbes made to order. *Science*, 303(5655):158, 2004.

[19] Mario Juhas, Leo Eberl, and John I Glass. Essence of life: essential genes of minimal genomes. *Trends in cell biology*, 21(10):562–568, 2011.

[20] Arcady Mushegian. The minimal genome concept. *Current Opinion in Genetics and Development*, 9(6):709–714, 1999.

[21] Bong Hyun Sung, Donghui Choe, Sun Chang Kim, and Byung-Kwan Cho. Construction of a minimal genome as a chassis for synthetic biology. *Essays in Biochemistry*, 60(4):337–346, 2016.

[22] Mitsuhiro Itaya. An estimation of minimal genome size required for life. *FEBS letters*, 362(3):257–260, 1995.

[23] Mario Juhas, Leo Eberl, and George M Church. Essential genes as antimicrobial targets and cornerstones of synthetic biology. *Trends in biotechnology*, 30(11):601–607, 2012.

[24] D Ewen Cameron, Caleb J Bashor, and James J Collins. A brief history of synthetic biology. *Nature Reviews Microbiology*, 12(5):381–390, 2014.

[25] George M Church, Michael B Elowitz, Christina D Smolke, Christopher A Voigt, and Ron Weiss. Realizing the potential of synthetic biology. *Nature Reviews Molecular Cell Biology*, 15(4):289–294, 2014.

[26] Adi L Tarca, Vincent J Carey, Xue-wen Chen, Roberto Romero, and Sorin Drăghici. Machine learning and its applications to biology. *PLoS Comput Biol*, 3(6):e116, 2007.

[27] Matthias Heinemann and Sven Panke. Synthetic biology—putting engineering into biology. *Bioinformatics*, 22(22):2790–2799, 2006.

[28] Eugene V Koonin. How many genes can make a cell: the minimal-gene-set concept. *Annual review of genomics and human genetics*, 1(1):99–116, 2000.

[29] Ren Zhang, Hong-Yu Ou, and Chun-Ting Zhang. DEG: a database of essential genes. *Nucleic acids research*, 32(suppl_1):D271–D272, 2004.

[30] Hao Luo, Yan Lin, Feng Gao, Chun-Ting Zhang, and Ren Zhang. DEG 10, an update of the database of essential genes that includes both protein-coding genes and noncoding genomic elements. *Nucleic acids research*, 42(D1):D574–D580, 2014.

[31] Ren Zhang and Yan Lin. DEG 5.0, a database of essential genes in both prokaryotes and eukaryotes. *Nucleic Acids Research*, 37(Database issue):D455–D458, 2009.

[32] Wei-Hua Chen, Pablo Minguez, Martin J Lercher, and Peer Bork. OGEE: an online gene essentiality database. *Nucleic Acids Research*, 40(D1):D901–D906, nov 2011.

[33] Wei-Hua Chen, Guanting Lu, Xiao Chen, Xing-Ming Zhao, and Peer Bork. OGEE v2: an update of the online gene essentiality database with special focus on differentially essential genes in human cancer cell lines. *Nucleic Acids Research*, 45(D1):D940–D944, oct 2016.

[34] Alison F Chalker and R Dwayne Lunsford. Rational identification of new antibacterial drug targets that are essential for viability using a genomics-based approach. *Pharmacology & therapeutics*, 95(1):1–20, 2002.

[35] John I Glass, Nacyra Assad-Garcia, Nina Alperovich, Shibu Yooseph, Matthew R Lewis, Mahir Maruf, Clyde A Hutchison, Hamilton O Smith, and J Craig Venter. Essential genes of a minimal bacterium. *Proceedings of the National Academy of Sciences*, 103(2):425–430, 2006.

[36] Iam Cooper and Ml Duffield. The in silico prediction of bacterial essential genes. *Science Against Microbial Pathogens*, (3):551–559, 2011.

[37] Leonard Guarente. Synthetic enhancement in gene interaction: a genetic tool come of age. *Trends in Genetics*, 9(10):362–366, 1993.

[38] Peter Novick, Barbara C Osmond, and David Botstein. Suppressors of yeast actin mutations. *Genetics*, 121(4):659–674, 1989.

[39] Jingyuan Deng. A Statistical Framework for Improving Genomic Annotations of Transposon Mutagenesis (TM) Assigned Essential Genes. In *Gene Essentiality*, pages 153–165. Springer, 2015.

[40] Jingyuan Deng, Lei Deng, Shengchang Su, Minlu Zhang, Xiaodong Lin, Lan Wei, Ali A Minai, Daniel J Hassett, and Long J Lu. Investigating the predictability of essential genes across distantly related organisms using an integrative approach. *Nucleic acids research*, 39(3):795–807, 2011.

[41] Jian Cheng, Zhao Xu, Wenwu Wu, Li Zhao, Xiangchen Li, Yanlin Liu, and Shiheng Tao. Training set selection for the prediction of essential genes. *PLoS ONE*, 9(1), 2014.

[42] Yu Chen and Dong Xu. Understanding protein dispensability through machine-learning analysis of high-throughput data. *Bioinformatics*, 21(5):575–581, 2005.

[43] Jingyuan Deng. An integrated machine-learning model to predict prokaryotic essential genes. In *Gene Essentiality*, pages 137–151. Springer, 2015.

[44] Kitiporn Plaimas, Roland Eils, and Rainer König. Identifying essential genes in bacterial metabolic networks with machine learning methods. *BMC systems biology*, 4(1):1–16, 2010.

[45] Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.

[46] Chuan Dong, Yan Ting Jin, Hong Li Hua, Qing Feng Wen, Sen Luo, Wen Xin Zheng, and Feng Biao Guo. Comprehensive review of the identification of essential genes using computational methods: Focusing on feature implementation and assessment. *Briefings in Bioinformatics*, 21(1):171–181, 2018.

[47] Xue Zhang, Marcio Luis Acencio, and Ney Lemke. Predicting essential genes and proteins based on machine learning and network topological features: a comprehensive review. *Frontiers in physiology*, 7:75, 2016.

[48] Ming Fang, Xiujuan Lei* Guo, and Ling. A Survey on Computational Methods for Essential Proteins and Genes Prediction, 2019.

[49] Chong Peng, Yan Lin, Hao Luo, and Feng Gao. A comprehensive overview of online resources to identify and predict bacterial essential genes. *Frontiers in Microbiology*, 8(NOV):1–13, 2017.

[50] João Paulo Müller da Silva, Marcio Luis Acencio, José Carlos Merino Mombach, Renata Vieira, José Camargo da Silva, Ney Lemke, and Marialva Sinigaglia. In silico network topology-based prediction of gene essentiality. *Physica A: Statistical Mechanics and its Applications*, 387(4):1049–1055, 2008.

[51] Soma Saha and Steffen Heber. In silico prediction of yeast deletion phenotypes. *Genet Mol Res*, 5(1):224–232, 2006.

[52] Yuan Yuan, Yanxun Xu, Jianfeng Xu, Robyn L Ball, and Han Liang. Predicting the lethal phenotype of the knockout mouse by integrating comprehensive genomic data. *Bioinformatics*, 28(9):1246–1252, 2012.

[53] Jian Cheng, Wenwu Wu, Yinwen Zhang, Xiangchen Li, Xiaoqian Jiang, Gehong Wei, and Shiheng Tao. A new computational strategy for predicting essential genes. *BMC Genomics*, 14(1), 2013.

[54] Kitiporn Plaimas, Jan-Phillip Mallm, Marcus Oswald, Fabian Svara, Victor Sourjik, Roland Eils, and Rainer König. Machine learning based analyses on metabolic networks supports high-throughput knockout screens. *BMC systems biology*, 2(1):67, 2008.

[55] Aditya Narayan Sarangi, Mohtashim Lohani, and Rakesh Aggarwal. Prediction of essential proteins in prokaryotes by incorporating various physico-chemical features into the general form of Chou's pseudo amino acid composition. *Protein and Peptide Letters*, 20(7):781–795, 2013.

[56] L W Ning, H Lin, H Ding, J Huang, N Rao, and F B Guo. Predicting bacterial essential genes using only sequence composition information. *Genet. Mol. Res*, 13(2):4564–4572, 2014.

[57] Jianxin Wang, Wei Peng, and Fang-Xiang Wu. Computational approaches to predicting essential proteins: A survey. *PROTEOMICS – Clinical Applications*, 7(1-2):181–192, jan 2013.

[58] Fredrick M. Mobegi, Aldert Zomer, Marien I. de Jonge, and Sacha A.F.T. van Hijum. Advances and perspectives in computational prediction of microbial gene essentiality. *Briefings in Functional Genomics*, 16(2):70–79, 2017.

[59] Michael Seringhaus, Alberto Paccanaro, Anthony Borneman, Michael Snyder, and Mark Gerstein. Predicting essential genes in fungal genomes. *Genome research*, 16(9):1126–1135, 2006.

[60] I King Jordan, Igor B Rogozin, Yuri I Wolf, and Eugene V Koonin. Essential genes are more evolutionarily conserved than are nonessential genes in bacteria. *Genome research*, 12(6):962–968, 2002.

[61] Yan Lin, Feng Gao, and Chun-Ting Zhang. Functionality of essential genes drives gene strand-bias in bacterial genomes. *Biochemical and biophysical research communications*, 396(2):472–476, 2010.

[62] Eduardo P C Rocha and Antoine Danchin. Essentiality, not expressiveness, drives gene-strand bias in bacteria. *Nature genetics*, 34(4):377–378, 2003.

[63] David J Lipman, Alexander Souvorov, Eugene V Koonin, Anna R Panchenko, and Tatiana A Tatusova. The relationship of protein conservation and sequence length. *BMC evolutionary biology*, 2(1):20, 2002.

[64] Xiaodong Gong, Shaohua Fan, Amy Bilderbeck, Mingkun Li, Hongxia Pang, and Shiheng Tao. Comparative analysis of essential genes and nonessential genes in Escherichia coli K12. *Molecular Genetics and Genomics*, 279(1):87–94, 2008.

[65] Zhenglong Gu, Lars M Steinmetz, Xun Gu, Curt Scharfe, Ronald W Davis, and Wen-Hsiung Li. Role of duplicate genes in genetic robustness against null mutations. *Nature*, 421(6918):63–66, 2003.

[66] Marcio L. Acencio and Ney Lemke. Towards the prediction of essential genes by integration of network topology, cellular localization and biological process information. *BMC Bioinformatics*, 10:290, 2009.

[67] Ernesto Estrada. Virtual identification of essential proteins within the protein interaction network of yeast. *Proteomics*, 6(1):35–40, 2006.

[68] Yih Chii Hwang, Chen Ching Lin, Jen Yun Chang, Hirotada Mori, Hsueh Fen Juan, and Hsuan Cheng Huang. Predicting essential genes based on network and sequence analysis. *Molecular BioSystems*, 5(12):1672–1678, 2009.

[69] Ronald Jansen, Dov Greenbaum, and Mark Gerstein. Relating whole-genome expression data with protein-protein interactions. *Genome research*, 12(1):37–46, 2002.

[70] Adam M Gustafson, Evan S Snitkin, Stephen C J Parker, Charles DeLisi, and Simon Kasif. Towards the identification of essential genes using targeted genome sequencing and comparative analysis. *Bmc Genomics*, 7(1):1–16, 2006.

[71] Yao Lu, Jingyuan Deng, Matthew B Carson, Hui Lu, and Long J Lu. Computational methods for the prediction of microbial essential genes. *Current Bioinformatics*, 9(2):89–101, 2014.

[72] Robi Polikar. Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6(3):21–45, 2006.

[73] Thomas G Dietterich. Ensemble Methods in Machine Learning. *International Workshop on multiple classifier systems*, pages 1–15, 2000.

[74] Louisa Lam and S Y Suen. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 27(5):553–568, 1997.

[75] Fumitaka Kimura and Malayappan Shridhar. Handwritten numerical recognition based on multiple algorithms. *Pattern recognition*, 24(10):969–983, 1991.

[76] Yu-Dong Cai, Lin Lu, Lei Chen, and Jian-Feng He. Predicting subcellular location of proteins using integrated-algorithm method. *Molecular diversity*, 14(3):551–558, 2010.

[77] Qianwu Ni and Lei Chen. A feature and algorithm selection method for improving the prediction of protein structural class. *Combinatorial chemistry & high throughput screening*, 20(7):612–621, 2017.

[78] Krishnaveni Palaniappan and Sumitra Mukherjee. Predicting "essential" genes across microbial genomes: A machine learning approach. *Proceedings - 10th International Conference on Machine Learning and Applications, ICMLA 2011*, 2:189–194, 2011.

[79] Jingyuan Deng, Lirong Tan, Xiaodong Lin, Yao Lu, and Long J. Lu. Exploring the optimal strategy to predict essential genes in microbes. *Biomolecules*, 2(1):1–22, 2012.

[80] Xiao Liu, Bao-Jin Wang, Luo Xu, Hong-Ling Tang, and Guo-Qing Xu. Selection of key sequence-based features for prediction of essential genes in 31 diverse bacterial species. *PloS one*, 12(3):e0174638, 2017.

[81] Yan Lin, Fa-Zhan Zhang, Kai Xue, Yi-Zhou Gao, and Feng-Biao Guo. Identifying bacterial essential genes based on a feature-integrated method. *IEEE/ACM transactions on computational biology and bioinformatics*, 2017.

[82] Arcady R Mushegian and Eugene V Koonin. A minimal gene set for cellular life derived by comparison of complete bacterial genomes. *Proceedings of the National Academy of Sciences*, 93(19):10268–10273, 1996.

[83] Robert E Bruccoleri, Thomas J Dougherty, and Daniel B Davison. Concordance analysis of microbial genomes. *Nucleic acids research*, 26(19):4482–4486, 1998.

[84] Rosario Gil, Francisco J Silva, Juli Peretó, and Andrés Moya. Determination of the core of a minimal bacterial gene set. *Microbiology and Molecular Biology Reviews*, 68(3):518–537, 2004.

[85] Hawoong Jeong, Sean P Mason, A-L Barabási, and Zoltan N Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, 2001.

[86] Yao Lu, Jingyuan Deng, Judith C. Rhodes, Hui Lu, and Long Jason Lu. Predicting essential genes for identifying potential drug targets in Aspergillus fumigatus. *Computational Biology and Chemistry*, 50:29–40, 2014.

[87] Jan Schellenberger, Richard Que, Ronan M T Fleming, Ines Thiele, Jeffrey D Orth, Adam M Feist, Daniel C Zielinski, Aarash Bordbar, Nathan E Lewis, and Sorena Rahmanian. Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2. 0. *Nature protocols*, 6(9):1290, 2011.

[88] Christof Francke, Roland J Siezen, and Bas Teusink. Reconstructing the metabolic network of a bacterium from its genome. *Trends in microbiology*, 13(11):550–558, 2005.

[89] Jeffrey D Orth, Ines Thiele, and Bernhard Ø Palsson. What is flux balance analysis? *Nature biotechnology*, 28(3):245–248, 2010.

[90] Wei Peng, Jianxin Wang, Weiping Wang, Qing Liu, Fang Xiang Wu, and Yi Pan. Iteration method for predicting essential proteins based on orthology and protein-protein interaction networks. *BMC Systems Biology*, 6, 2012.

[91] Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, and Shanrong Zhao. Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, 18(6):463–477, 2019.

[92] NCBI Resource Coordinators. Database resources of the national center for biotechnology information. *Nucleic acids research*, 44(D1):D7–D19, 2016.

[93] Dennis A Benson, Ilene Karsch-Mizrachi, David J Lipman, James Ostell, and David L Wheeler. GenBank. *Nucleic Acids Research*, 35(suppl_1):D21———D25, dec 2006.

[94] John F Peden. Analysis of codon usage. 2000.

[95] Anders Krogh, Björn Larsson, Gunnar Von Heijne, and Erik L L Sonnhammer. Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *Journal of molecular biology*, 305(3):567–580, 2001.

[96] Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L Madden. BLAST+: architecture and applications. *BMC bioinformatics*, 10(1):421, 2009.

[97] Nancy Y Yu, James R Wagner, Matthew R Laird, Gabor Melli, Sébastien Rey, Raymond Lo, Phuong Dao, S Cenk Sahinalp, Martin Ester, and Leonard J Foster. PSORTb 3.0: improved protein subcellular localization prediction with refined localization subcategories and predictive capabilities for all prokaryotes. *Bioinformatics*, 26(13):1608–1615, 2010.

[98] Peter Rice, Ian Longden, and Alan Bleasby. EMBOSS: the European molecular biology open software suite, 2000.

[99] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, and Vincent Dubourg. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[100] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, and Jonathan Bright. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods*, 17(3):261–272, 2020.

[101] Mark Andrew Hall. Correlation-based feature selection for machine learning. 1999.

[102] J Peden. CodonW. *Trinity College*, 1997.

[103] Ruth Hershberg and Dmitri A Petrov. General rules for optimal codon choice. *PLoS Genet*, 5(7):e1000556, 2009.

[104] Jeffrey L Bennetzen and Benjamin D Hall. Codon selection in yeast. *Journal of Biological Chemistry*, 257(6):3026–3031, 1982.

[105] Francisco Melo. Area under the ROC Curve BT - Encyclopedia of Systems Biology. pages 38–39. Springer New York, New York, NY, 2013.

[106] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[107] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[108] Manoranjan Dash and Huan Liu. Feature selection for classification. *Intelligent data analysis*, 1(3):131–156, 1997.

[109] Sebastian Raschka. MLxtend: providing machine learning and data science utilities and extensions to Python's scientific computing stack. *Journal of open source software*, 3(24):638, 2018.

[110] Luo Xu, Zhirui Guo, and Xiao Liu. Prediction of essential genes in prokaryote based on artificial neural network. *Genes and Genomics*, 42(1):97–106, 2020.

[111] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[112] Hong Li Hua, Fa Zhan Zhang, Abraham Alemayehu Labena, Chuan Dong, Yan Ting Jin, and Feng Biao Guo. An approach for predicting essential genes using multiple homology mapping and machine learning algorithms. *BioMed Research International*, 2016, 2016.

[113] Karthik Azhagesan, Balaraman Ravindran, and Karthik Raman. Network-based features enable prediction of essential genes across diverse organisms. *PLoS ONE*, 13(12):1–13, 2018.

[114] Jesse Davis and Mark Goadrich. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.