The University of
Nottingham

A Random Forest Approach to Understanding CRISPR-Cas Associations in Bacteria

Kyle Jamie Millar

Master’s Thesis in Bioinformatics
Faculty of Life Sciences
University of Nottingham

Supervisor	Professor James McInerney and Professor Mary O’Connell
Co-supervisor	Dr Jasmine Ono
External Examiner	Professor Chris Creevey

24th June 2025

Abstract

CRISPR-Cas systems are a crucial and intriguing defence mechanism found in bacteria and archaea. This defence mechanism is able to adapt and defend against attacks from Mobile Genetic Elements (MGEs). This mechanism also has many uses outside of genomic defence. For example, certain types of CRISPR-Cas proteins allow for modification of eukaryotic genomes in vivo. Currently, there are applications and algorithms capable of finding CRISPR-Cas types and the associated arrays, however, the idea of predicting whether a genome might contain a CRISPR-Cas locus, based purely on the background genome content offers a faster query time. To test whether the presence of CRISPR-Cas systems could be predicted from the background genome, a Random Forest algorithm was employed using a large data set - a bacterial pangenome containing 9,689 genomes. To annotate this pangenome with 'CRISPR' identifiers, the annotation tool Bakta was used, allowing for the use of custom scripts to find the relevant information needed from the annotated genomes. The algorithm was shown to have an accuracy of 0.89, and an AUC-ROC score of 0.96. These results imply a strong ability to classify the predictions correctly, based on background genome content. The algorithm calculated the 'feature importance' of all genes that were present in the pangenome; the gene of highest importance was 'pbp4b' followed closely by 'csy3' (a positive control variable). The ten genes that had the highest feature importance all had a statistically significant association with CRISPR-Cas systems when evaluated using chi-squared tests. The algorithm was capable of predicting CRISPR-Cas systems in γ-proteobacteria and offers potential for research candidates when investigating CRISPR-Cas associations. This approach could be used to predict CRISPR-Cas more broadly across prokaryotic life, upon data availability.

Acknowledgements

I would like to acknowledge Professor James McInerney, Professor Mary O'Connell and Dr Jasmine Ono for their excellent supervision and support throughout this project. Both the McInerney and O'Connell lab members offered support whilst carrying out this project. Finally, I acknowledge the high-performance computing system Ada and the supporting staff from the Ada IT team for enabling the research project.

Contents

1 Abbreviations xii

1 Introduction 1

1.1 CRISPR-Cas systems 4

1.2 Types of CRISPR 6

1.3 Random Forests 7

1.4 Protein Systems and Co-evolution 10

1.5 Other Defence Mechanisms 11

1.6 Internal Genetic Defence Systems 12

1.7 Membranome 12

1.8 Genome annotation and Pangenome Creation 13

1.9 Why γ-proteobacteria? 14

2 Materials And Methods 17

2.1 Genome Acquisition 19

2.1.1 Moving Files with Specific File Endings 19

2.2 Genome Annotation using Bakta 19

2.2.1 Creating List of Files to be Processed 20

2.2.2 Running Bakta Process 20

2.3 Pangenome Construction 20

2.3.1 Moving 1000 Files at a Time 20

2.3.2 PanTA Code Edited: (main_pipeline.py) 21

2.3.3 Binarizing Dataset 22

2.4 Removal of Extraneous Data Points 22

2.4.1 Counting to Calculate 22

2.5 Annotating Pangenome With CRISPR Data Points 23

2.6 Removing CRISPR Associated Genes 24

2.7 Random Forest Algorithm 24

2.7.1 Code used for Random Forest Algorithm 24

2.8 Creating Contingency Tables from the Pangenome 26

2.8.1 Code Used to Construct Contingency Tables 26

3 Results **29**
3.1 Preprocessing 29
3.1.1 Removal of Genes Not Found in More Than 2% of Genomes 29
3.1.2 Excerpt of First Pangenome 30
3.1.3 Pangenome Gene Category Statistics 31
3.1.4 Excerpt of Pangenome After Binarization 32
3.1.5 CRISPR Analysis of Pangenome 33
3.1.6 Calculated Accuracy Metrics 34
3.1.7 Algorithm Calculated Feature Importances 35
3.1.8 Significance for Top 10 Genes Within the Pangenome . 37
3.1.9 Significance of 'pbp' Homologous Family Genes Within the Pangenome 39

4 Discussion **41**
4.1 Analysis of 'PBP4B' 43
4.2 Other Gene Analysis 45
4.2.1 The Gene 'ais' 45
4.2.2 The Gene 'lsrA' 46

5 Conclusion **49**

Appendix **51**

Bibliography **53**

List of Figures

1.1 CRISPR-Cas system overview. 4

2.1 Pipeline Overview 18

4.1 StringDB gene network of pbp4b 44
4.2 StringDB gene network of ais 46
4.3 StringDB gene network of lsrA. 47

List of Tables

3.1 Excerpt of Pangenome. 30
3.2 Pangenome Gene Category Statistics. 31
3.3 Excerpt of Pangenome After Binarization. 32
3.4 Excerpt of the Pangenome After CRISPR Identification. 33
3.5 Calculated Accuracy, Precision, Recall, F1 Score and AUC-ROC . 34
3.6 Feature Importance Table. 35
3.7 Contingency tables and chi-squared tests for the top 10 genes removing the positive controls 36
3.8 Contingency tables and chi-squared tests for Penicillin binding protein families 38

1 ABBREVIATIONS

AMR : Antimicrobial Resistance

AUC-ROC : Area Under the Receiver Operating Characteristic Curve

Cas : CRISPR associated genes

CRISPR : Clustered Regularly Interspaced Short Palindromic Repeats

CSV : Comma-Separated Values (file type)

DNA : Deoxyribose Nucleic Acid

GFF3 : Generic Feature Format version 3 (file type)

HGT : Horizontal Gene Transfer

MGE : Mobile Genetic Element

ML : Machine Learning

NCBI : National Centre for Biotechnology Information

RF : Random Forests (machine learning model)

Sklearn : Sci Kit learn

TSV : Tab Separated Values (file type)

1 Introduction

The aim of this research project is to find new Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR) associated genes (Cas), within the genomes of bacteria. An "associated gene" is one whose presence or absence in a genome appears to be modulated by the presence or absence of CRISPR-Cas elements. With the enormous expansion in the availability of prokaryotic genomes, we can now use sophisticated computational approaches to fully understand the extent to which the presence or absence of a gene is potentiated by the presence or absence of CRISPR-Cas.

Traditionally, understanding the factors that influence CRISPR-Cas presence and function in bacteria requires the analysis of complex, genomic and ecological data. Historical research methods, while useful for identifying linear correlations, may struggle to capture the intricate interactions and non-linear relationships that characterize bacterial genomic evolution. Machine learning (ML) offers a modern alternative by enabling data-driven pattern recognition and predictive modelling, making it particularly suitable for genomic studies where relationships between variables are not always obvious. New methods of analysis find that ML has emerged as a powerful tool for genomic analysis, capable of identifying complex, non-linear relationships within large datasets (Monaco et al., 2021) . In research, ML methods have been applied to classify system subtypes (Russel et al., 2020) and explore phage-host interactions (Dimitri Boeckaerts et al., 2024). Among ML techniques, the Random Forest (RF) algorithm is particularly well-suited for this study due to its ability to handle large datasets with numerous predictor variables while maintaining high accuracy and interpretability (Breiman, 2001) . Applying a RF algorithm facilitates the use of a diverse dataset to explore these associations.

Additionally, this project aimed to create a functional algorithm to predict whether a bacterium will utilise CRISPR-Cas systems in their defensome. The defensome of a bacterium is defined as all of the systems and genes which are used in the defence of a cell, a prokaryotic immune system (Goldstone et al., 2006; Beavogui et al., 2024). This approach could enable the use of the bacterial transcriptome, a collection of all of the RNA sequences which are being transcribed within the cell at the time of sampling (Wang, Gerstein and

Snyder, 2009), or the fully annotated genome to reliably predict this system in clinical settings. Future work could therefore lead to the prediction of whether a bacterial infection is amenable to the use of phage therapy. The development of Phage therapies takes time (Leptihn and Loh, 2022) and if the bacterium can use CRISPR-Cas to defend from the chosen phage, the therapy would be ineffective.

This study does not only apply to CRISPR-Cas systems; it could also be used to find associations or correlations with other biological systems or functions. Feasible uses for this methodology lie within AMR and gene studies where there are more associations to be found. The use cases for ML in bioinformatics are endless, as it widens opportunities for large amounts of information to be identified and analysed. A challenge with ML is its heavy dependence on large volumes of data to be efficient and accurate.

Understanding CRISPR-Cas systems' associations within bacterial genomes could assist us in understanding which genes and type of environment makes CRISPR-Cas systems efficient. What types of genetic spacers do these systems acquire? Which genes inhibit or directly impact the capabilities or accuracy of these systems? These are questions that must be answered before attempting to implement this kind of gene therapy for use in eukaryotes. Currently, the use of CRISPR-Cas9 in functional genomics, specifically to edit gene variants and introduce new genes into genomes, is revolutionary (Kim, Kweon and Kim, 2024) . This application allows for the comprehension of how different elements of these genes interact within the studied organism (Kim, Kweon and Kim, 2024). Historically, functional genomic insight was mainly based on the study of naturally occurring genetic mutations, however now, the use of CRISPR-Cas9 systems enables more research and development within this field (Agrotis and Ketteler, 2015) .

Bacteria have a cosmopolitan global distribution, being found in every ecological niche on the planet, from soil to the rumen and guts of animals (Ahmed and McKay, 2024). Bacteria are a significant ecological driving force in all environments and ecosystems of which they are a part (Ahmed and McKay, 2024). They are also the etiological agents of many infections and diseases. The adaptable nature of bacteria allows them to persist in numerous environments; this often means that bacteria have become pathogenic and can predate on humans and other organisms. Infectious diseases are on the rise and the ability of frontline antibiotics to deal with infection is diminishing (Salam et al., 2023). Antimicrobial resistance (AMR) is becoming more problematic in clinical settings. The overprescription or the widespread use of antibiotics within the meat humans eat or the food we feed our livestock could be at fault for AMR (Bava et al., 2024). Nonetheless, the discovery of antibiotics may have been one of the most integral discoveries of the 20th century, allowing humans to increase their lifespan and have a capability to treat bacterial infections (Nicolaou and Rigol, 2017).

Bacteria can be manipulated to assist us in certain practices and can be genetically modified to produce numerous substances. A major use of bacteria is in bioremediation, the employment of bacteria to breakdown or degrade contaminants in soil and water (Gupta and Gandhi, 2023). One specific example is *Deinococcus radiodurans*, an extremely radioresistant organism that has been genetically engineered to assist in the removal of toluene and ionic mercury from nuclear waste (Vaishnav and Demain, 2009). Though bacteria can often be a hindrance in modern medicine, they can also provide innovative solutions in infectious disease prevention, with *Pseudomonas aeruginosa* and *Rhodopseudomonas capsulata* being used to produce gold nanoparticles for different therapeutic purposes (Singh and Kundu, 2013).

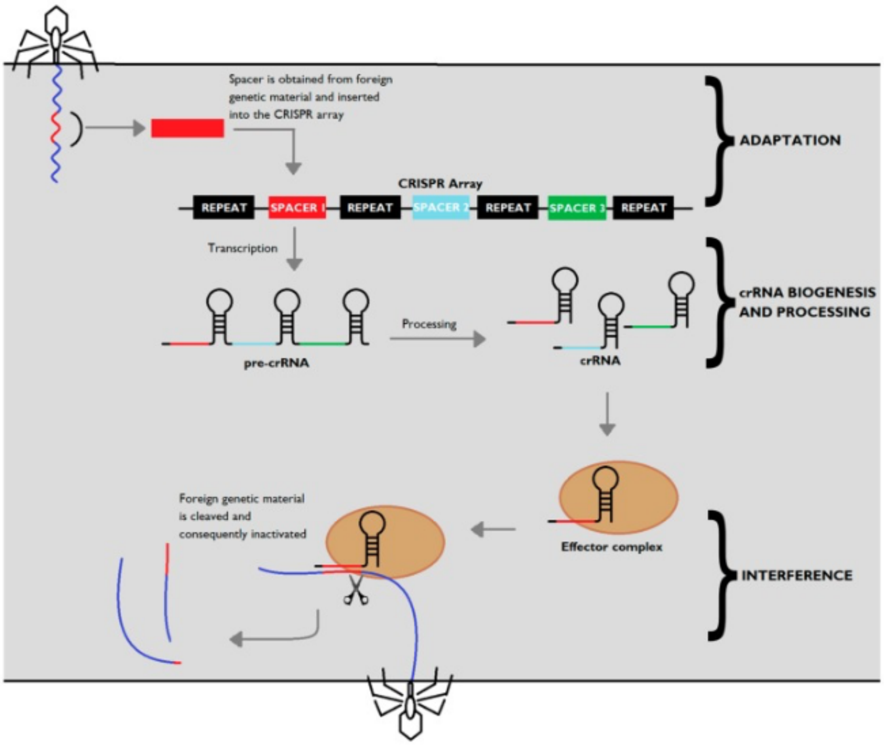
With the rise of AMR bacterial strains there has been a greater urgency in the field of clinical science to find new therapeutic responses. Research carried out by Sawa, Moriyama and Kinoshita (2024) highlights a key opportunity that humans could use to their advantage in this war against AMR: phage therapy. This approach has shown promise in the effort to manage AMR bacteria. Bacteriophage offer a feasible way to infect specific bacteria that are resistant to the other therapeutic options (Łusiak-Szelachowska et al., 2022). That said, to use bacteriophage to attack a bacterium we must understand the specific defence mechanisms of the bacteria, which have been evolving and adapting for billions of years (Abedon, 2012). Bacteria have evolved many different defence mechanisms that can be used in diverse ways to protect themselves from attack.

While this project focusses on uncovering associations between CRISPR-Cas systems and the rest of the genome, there are uses for CRISPR-Cas9 outside of the bacteria in which this system is found. In 2012, Jinek et al. published ‘A Programmable Dual-RNA–Guided DNA Endonuclease in Adaptive Bacterial Immunity’ which marked a new opportunity in research regarding gene editing and genetic engineering in eukaryotic life using the CRISPR-Cas9 protein. This research explored the capabilities that the Cas9 protein could have, whilst used in conjunction with modern techniques of synthesizing genetic material (Jinek et al., 2012). Fixing defective genes, replacing cancerous genetic markers, and alleviating hereditary diseases are among some of the capabilities that the use of Cas9 led gene editing could pose for medical use in humans (Zhang and McCarty, 2016). However, a few issues arise when attempting to use this incredible discovery within such complex genomes. In the process of utilising CRISPR-Cas9 proteins in eukaryotic life, it is critical that the risks associated with off-target cleavage, which can lead to unintended and volatile consequences, are understood (Ochiai and Yamamoto, 2023).

1.1 CRISPR-CAS SYSTEMS

CRISPR-Cas is a sophisticated biological defence system that is used by almost 40% of all bacterial genomes and 80% of all archaeal genomes (Zink, Wimmer and Schleper, 2020). These defence mechanisms are used to protect the genome from genetic attack. A notable difference between CRISPR-Cas and Restriction-Modification (R-M) systems is that CRISPR-Cas systems can adapt, such that the cell becomes immune to attack from specific external threats. Briefly, the CRISPR-Cas system works by assimilating a spacer of foreign DNA into the CRISPR array, transcribing the array to form a pre-crRNA complex. This pre-crRNA complex is then processed into an individual crRNA unit which is then implemented into the effector complex, allowing the system to intercept the incoming foreign DNA and cleave it into inactivated strains of DNA (Figure 1.1)

Figure 1.1: CRISPR-Cas system overview.
The figure outlines the three major phases of defence – adaptation, crRNA Biogenesis and processing, and Interference. Used with permission from Prof. Gabriela Jorge da Silva (Loureiro and da Silva, 2019) .


Spacer is obtained from foreign genetic material and inserted into the CRISPR array
ADAPTATION
CRISPR Array
REPEAT
SPACER 1
REPEAT
SPACER 2
REPEAT
SPACER 3
REPEAT
Transcription
Processing
pre-crRNA
crRNA
crRNA BIOGENESIS AND PROCESSING
Foreign genetic material is cleaved and consequently inactivated
Effector complex
INTERFERENCE

This system was initially mentioned by Ishino et al. (1987) and later characterized by Jansen et al. (2002). The uses of the defensive abilities are vast and capable; researchers found that the CRISPR-Cas system can defend from Bacteriophage attack (Marraffini and Sontheimer, 2008) ; this form of attack is common throughout a bacteria's life span. CRISPR-Cas allows for the identification, recognition and destruction of the genetic material that has been introduced into the organism (Loureiro and da Silva, 2019). Horizontal gene transfer (HGT), the movement and transfer of genes between organisms, presents an interesting case where the CRISPR-Cas system is employed to recognise and inhibit the process. There are reported cases of HGT being limited through the application of CRISPR-Cas to protect the genome (Wheatley and MacLean, 2020). Conversely, there has been research which contradicts these reported cases where no inhibitions of HGT could be proven (Gophna et al., 2015). There are many reasons why CRISPR-Cas is used in this manner, but the most interesting hypothesis is that the stability of the genome, and henceforth the survival of the organism, could be adversely affected by the introduction of new genes through HGT (Marraffini and Sontheimer, 2008). The requirement for the system to recognise the invading DNA sequence and respond specifically to it causes the systems slow reaction; both the expression of the relevant spacer sequence and its use to neutralise the invading DNA can take time (McKenzie et al., 2022). The other limitation of the CRISPR-Cas defence system is that it requires many proteins to be expressed. The assembly of these proteins limits the efficiency of the system and requires considerable resources to be poured into it (Zaayman and Wheatley, 2022) . The efficiency trade-off associated with the CRISPR-Cas system may explain why 60% of organisms that could potentially utilise it do not retain it in their genomes (Jiang et al., 2013). In some cases, this could be the result of natural selection, such that maintaining the system imposes a fitness cost greater than its benefit. Alternatively, genetic drift may lead to the loss of the CRISPR-Cas system in populations where it provides no significant advantage, rendering it effectively neutral over evolutionary time. This system would confer a significant deleterious effect in an environment where multiple bacteriophage attacks were taking place simultaneously. It is easily speculated that this lack of efficiency may be the reason that this system would be deleted from a genome (Hille and Charpentier, 2016).

CRISPR-Cas systems are suggested to have evolved from integrases, which are elements similar to transposons (Koonin and Makarova, 2019); these elements are responsible for integration of spacers into CRISPR arrays, and this relationship suggests that the adaptability of CRISPR-Cas systems may have originated from these elements. The integrated genetic material is used as the (g)RNA within the system which enables the recognition and destruction of the invading genetic material (Koonin and Makarova, 2019).

1.2 TYPES OF CRISPR

There are many types of CRISPR-Cas systems which employ different unique genes and proteins in different combinations; significant differences between theses types enable distinct levels of defence. CRISPR-Cas types I, II and III all use different unique genes within their systems: Cas3 for type I, Cas9 in type II and finally Cas10 in type III. These types are found in many varied species across both Bacteria and Archaea (Makarova and Koonin, 2015)).

Some Cas proteins have been shown to have different efficiencies in gene editing scenarios. A study published by Banakar et al. (2020) compared two different Cas proteins, Cas9 and Cas12a, that cleave DNA in different ways. In situations where Cas9 would cleave to produce a blunt end of DNA, Cas12a cleaves to create a staggered DNA end. Although Cas genes have similar functions within the biological system, they vary in their efficiency at cleaving DNA and causing genetic disruption (Makarova et al., 2019) . These differences in efficiency may influence which organisms adopt certain CRISPR-Cas systems, depending on environmental pressures or the presence of other existing defence mechanisms (Makarova et al., 2019) . There is significant amounts of research carried out on Cas9 proteins, which have various reports of Cas9 cleaving with blunt ends, 1-base pair staggered ends or multiple base pair staggered ends (Stephenson, Raper and Suo, 2018). This is a key concern regarding the capabilities of this type of gene editing that uses Cas proteins in vivo. The variation in results reported implies that Cas9 can cut in multiple ways, however, this variation has yet to be explained in published research.

The types of CRISPR-Cas systems that are present across biological life differ in many ways. In this project we did not discriminate between CRISPR-Cas subtypes due to the challenges associated with labelling and sorting data in a project of this duration. The amount of data required to discriminate the subtypes would also have been a limiting factor in terms of having comparable, or balanced, numbers of each CRISPR-Cas type in order to prevent overfitting.

Phages and their respective bacteria have been in an arms race for billions of years (Koskella and Brockhurst, 2014). Some of these phages have evolved to have CRISPR-Cas aversion through the inhibition of the CRISPR-Cas proteins (Stanley and Maxwell, 2018). This presents an interesting opportunity to apply these phages within medicine. If the use of a designer phage is required for treatment of a bacterial infection, the use of this protective inhibition could be added to all phages to universally remove the chance of the bacteria defending from the phage attack. However, research carried out by Camara-Wilpert et al. (2023) found that the inhibition features of these ‘Racr’ proteins are specific to the type of CRISPR-Cas system it is trying to inhibit. This necessitates the sequencing of the bacterial genome to identify the type of CRISPR-Cas system present prior to deciding which inhibiting protein would work for that specific bacterium. The potential of these inhibition proteins is currently unrealised, though once fully characterised, these proteins could offer a deeper understanding of mechanisms to fight these dangerous infections.

1.3 RANDOM FORESTS

RF algorithms are a powerful ML methodology, commonly used for classification and regression tasks (Breiman, 2001). The algorithm operates through the creation of many decision trees and calculating the mode when classifying, or alternatively the mean prediction when working in regression, for each tree. To form the forest the algorithm uses decision trees, these trees split the data based off of the significance levels of the differentiators in the input features, creating a set of simple decision rules which are interpreted from the input data features (Breiman, 2001) . RF algorithms use a technique called 'bagging' which allows multiple subsets of the original data to be used to train trees on different pieces of data; this allows for diverse training of the algorithm. Random feature selection at each split of the tree increases the diversity of data features used to train the algorithm, enabling uncorrelated data features to be used in each tree. The algorithm then uses the decision trees it has created to 'vote' at the end of training; when the algorithm is classifying, the majority vote is selected, whereas when the algorithm is being used for regression, the average output of all the trees in the forest is computed (Breiman, 2001). These features of the model allow for use with Big Data and enables increased predictive performance when compared to stand alone decision trees.

'Big data' is a term used to define data sets which are orders of magnitude larger than datasets which can be analysed by conventional methods. This type of data set is linked with generation speed and volume of data which can both be affected by the types of data being used and the accuracy of said data (Greene et al., 2015). While being used for many diverse types of analysis in bioinformatics, it is also used in genomics, proteomics, transcriptomics, and more. In specific, the use of Next-Generation Sequencing (NGS) techniques, can quickly generate terabytes of data. Illumina, PacBio and Oxford Nanopore sequencing all require techniques to analyse the output of genomic data in a fast and efficient manner (Gupta, Kumar and Kumar, 2023). The techniques used for analysing this type of dataset require the capability to understand patterns. Common methodologies which use Big Data are ML models; the capabilities these models have for understanding and analysing copious amounts of varied data enables bioinformaticians to find patterns previously unrecognisable in the data (Greene et al., 2014). The combinations of data which can be used is endless within this space. The use of genomic and transcriptomic data by Curtis et al. (2012) has enabled the analysis of the data of 2000 breast tumours and has revealed novel subgroups. Some challenges which are faced when using Big Data in bioinformatics are the requirements for storage; high-performing computing facilities limit the capabilities of research to groups or companies which can afford to use these types of facilities. Another challenge regarding the use of Big Data is that currently, there is no standardised file format and structure for types of information available in online repositories and databases. Furthermore, data privacy and security must also be considered, as human genomic data is considered confidential information and would require ethical and legal understanding of what is required to protect this type of information(Greene et al., 2015). The future direction of Big Data in bioinformatics could include the possible use of quantum computing and a deeper use of artificial intelligence and ML models (Li et al., 2019).

There are a few advantages of using this model compared to other models. One advantage of this model is reducing overfitting; the use of multiple different trees allows the model to calculate using vast amounts of data, thus improving the capabilities of the algorithm regarding the diversity of inputs and decisions made from those inputs. Another advantage is the model's capability to handle Big Data, the model allows for use of large datasets, enabling use within complex tasks or datasets where associations are difficult to find. Another advantage is the model's stability with 'noisy data', when parts of the data are not associated or integral to the objective of the analysis the model handles the irrelevant data effectively.

For RF to be used appropriately with the data, we must understand the type of data required for RF - a pangenome is used in this project. The pangenome allows us to use it as a matrix to enable features to be used for decision trees. The matrix enables the algorithm to understand if a gene family is present or absent in the genome. These gene families are seen as features for use in the decision trees.

For this project we applied the ML architecture of RF to approach the project. An RF algorithm can be used for many different problems, it allows for use of randomised decision trees to calculate probabilities based off of a 'training set' and a 'test set', these sets are randomised. A predetermined percentage of the dataset will be used to train the algorithm, and the remaining data will then be used to test the algorithms capabilities. These algorithms are particularly useful in answering yes or no questions, or in other words binary questions. This lies in agreement with the current assumption of CRISPR-Cas systems; either an organism will have the genes within their genome or they will not. Using this architecture, we can feed large pangenomes into the algorithm and use the presence or absence of a gene to calculate a probability. Subsequently, we can find the associated genes with the CRISPR-Cas system.

This method allows for a large amount of data to be used to find the associations. There are many computer programming libraries available which contain a function to use RF. The one used within this project is Sklearn (Pedregosa et al., 2011), in the Python programming language (Python Software Foundation, 2024). This allowed for a quick turn around with development and allowed for fast customisation of each parameter. Specifically used for limiting RAM usage were: 'max_depth' which is the maximum depth each decision tree will reach before being stopped, 'n_estimators' which is the number of decision trees which the algorithm uses to calculate the feature importances, and finally 'n_jobs' which is the number of CPU cores which are used by the algorithm for parallelisation. The modification of these parameters is crucial to the effectiveness of the algorithm, increasing or decreasing accuracy and computational time accordingly. The Sklearn library also has features which can report the results of the algorithms calculations, these can be customised as well. Accuracy, Precision and Recall are all metrics that are calculated to understand the effectiveness of the algorithm, the equations used to calculate these are found below.

$$Accuracy = \frac{True\ Positive(TP) + True\ Negative(TN)}{True\ Positive(TP) + True\ Negative(TN) + False\ Positive(FP) + False\ Negative(FN)}$$

$$Prescision = \frac{True\ positive(TP)}{True\ Positive(TP) + False\ Positive(FP)}$$

$$Recall = \frac{True\ positive(TP)}{True\ positive(TP) + False\ Negative(FN)}$$

(Chen and Liaw, n.d.)

These equations use four key metrics: True Positives (TP), which are the correctly predicted positive cases; False Positives (FP), which are negative cases incorrectly classified as positive; True Negatives (TN), which are correctly predicted negative cases; and False Negatives (FN), which are positive cases incorrectly classified as negative. These values provide measurements of the performance of the ML algorithm. The strength of the associations within the dataset can affect these metrics. While a high accuracy (>0.9) is often considered strong, accuracy can be misleading if interpreted alone, especially in imbalanced datasets. Therefore, other metrics such as Precision, Recall, and the AUC-ROC score are important for evaluating performance. The AUC-ROC score reflects the model's ability to distinguish between classes across all classification thresholds (Çorbacıoğlu and Aksel, 2023). It is particularly useful for binary classifications, as it shows how well the model separates positive cases from negative cases regardless of the decision boundary. Given the binary nature of both the data and the classification task in this project, the AUC-ROC score is a metric which will be used for model evaluation. However, Precision, Recall, and Accuracy are also calculated to assess changes in hyper-parameters and how input data affects overall model performance.

1.4 PROTEIN SYSTEMS AND CO-EVOLUTION

Proteins rarely work in isolation, these molecules work in systems and pathways. These systems are commonly called networks and are found through associations computationally and experimentally (Spirin and Mirny, 2003). These Protein-Protein Interactions (PPIs) are multifunctional and capable of enabling adaptive responses to environmental changes and signals (Westermarck, Ivaska and Corthals, 2013), these PPIs allow cells to exhibit many diverse external features (Nooren, 2003) and fulfil many roles in cellular signalling, enabling communication and coordination between organisms of the same and different species (Nada et al., 2024). Certain proteins and genes may not be part of a specific pathway or system, but these proteins still interact and associate with various systems, increasing the efficiency of the system as a whole. These types of proteins are called 'moonlighting proteins' (Huberts and van der Klei, 2010). The multitasking capabilities that these molecules provide is essential to saving energy and enabling certain biological systems to perform their role efficiently. An example of this is the protein pyruvate kinase (PykA), it has been linked with replication, and research has implicated that this specific enzyme has the capability to interact and react to signalling molecules (Horemans et al., 2020). These proteins do not only interact with other proteins; some can be used to enable transcription of genes in specific scenarios or even inhibit transcription of proteins by way of binding to enhancers and promoters (Westermarck, Ivaska and Corthals, 2013).

The co-evolution of genes and their corresponding proteins has allowed adaptation and efficiency to be meticulously tested for survival fitness, elucidating the full extent of the roles each gene has within a network can allow for the understanding of why certain genes have evolved alongside others (Dilucca, Cimini and Giansanti, 2021). Research shows that certain genes which have similar patterns in codon usage are usually part of a PPI system together; these interactions have been conserved over generations due to the beneficial nature of the system, forming a lineage of co-evolution (Fraser et al., 2004). The co-evolution of transcription factors and the specific binding sites have evolved simultaneously alongside their specific genes to allow for tuning and control of the response to certain stimuli (Yang et al., 2011). These factors enabling the co-evolution of both proteins which interact in systems or PPIs and the transcriptional factor tuning, has allowed for associations to be formed outside of the system specific proteins. Understanding how certain genes interact or associate with genes outside of the systems which they are involved in is crucial to fully understand how the genome works. Understanding the fundamental associations could also shine a light on the selective capabilities which bacteria hold when being presented genes through Horizontal Gene Transfer (HGT; Yang et al., 2011). Why would a bacteria choose to have one gene over another if presented simultaneously? Is this just random chance or is this decision made through a selection process of associations between the proposed addition and the current genome? If the associations are not beneficial and only hinder the organism, this could explain why certain systems, as an example CRISPR-Cas systems, are not found within more of the population of bacteria.

1.5 OTHER DEFENCE MECHANISMS

Phages and bacteria have a relationship similar to that between viruses and humans, parasitic in nature. Phages invade the cell and inject their genomes into the bacteria to integrate and use the bacteria to replicate itself within the host organism (Huang et al., 2021). The phages which infect the bacteria are very specialised and specific to each bacterium. Being species-specific allows for the specialisation of infection methods. Binding to the cell surface membrane of the organism is the most direct route to injection of genetic material into the bacteria (Huang et al., 2021). Infection can be halted at the first interaction between the bacteria and phage by modification of the external proteins to which these phages bind. If the phage cannot bind to the cell surface membrane, they are unable to inject their genomes into the bacterial cell. This is a passive defence system which can be used by the bacteria to increase its survival until the phage mutates and is able to interact and bind to the newly formed protein (Wang, Fan and Tong, 2023) . Reconstructing external structures such as membrane lipids is the backbone of defence for the bacteria, however it can also affect the virulence of a bacteria. Changing the structure of capsular polysaccharides can reduce the phage susceptibility. This is just one example of extracellular structures that can be modified to defend the assault of the phages (Wang and Leptihn, 2024). If the binding of a terminal phage can be defended against, then the fitness of the bacterium is increased; simple changes of structure can come about by mutation of the genome. Sorensen et al. (2011) reported that phage F336, which infects *Campylobacter jejuni*, recognises the capsular phosphoramidate which is found on its cell surface. Phages binding to non-protein extracellular structures creates numerous opportunities for the subversion of the bacterial defence systems. Changing the structures of extra-cellular polysaccharides can affect the fitness of the bacteria within the environment, affecting their resistance to desiccation and their ability to adhere to surfaces in their environment (Bazaka et al., 2011). Superinfection exclusion systems are defence systems that function at the cell surface level, working to prevent the injection of phage DNA into the cytoplasm (Cumby et al., 2012). Research has showed how specific proteins which are bound to the inner membrane of the bacterial cell prevent the DNA of specific phage from entering the cell (Cumby et al., 2012); these proteins only defend the cell from the DNA of a specific phage. An efflux of potassium ions from cells into the growth medium correlates with the injection of phage DNA into the cell. When gp15, a mucin glycoprotein found in certain bacteriophages, was present there was no efflux measured when its specific phage HK97 attempted to inject the DNA into the cell. The results of the study imply that a bacteria can adaptively defend against the phages that are present in its natural environment by actively blocking foreign DNA from entering, using specific proteins that can combat infection by those bacteriophages.

1.6 INTERNAL GENETIC DEFENCE SYSTEMS

Microbes use sophisticated systems that have evolved and adapted over billions of years to defend against introgression by Mobile Genetic Elements (MGE) and phages. Common genetic defence systems include R-M systems, CRISPR and CRISPR-Cas genes. Bacteria can also defend the rest of the colony by causing self-destruction if infected by a phage for which there is no possible defence (Lopatina, Tal and Sorek, 2020). R-M systems are defence systems that are used to defend from genetic attacks; these systems are used by the bacteria to defend against specific MGE. These systems are non-adaptive and only defend against specific genetic sequences. They are separated into 4 types (I-IV; Vasu and Nagaraja, 2013). Each type consists of different genes, or more specifically, different combinations of genes. Most are found to contain a restriction enzyme (R unit), a modification enzyme (M unit) and a specificity unit (S unit) (Labrie, Samson and Moineau, 2010). These systems only identify specific DNA sequences if the invading genetic material does not contain the specific sequences targeted by the R-M system; this then leaves a gap in the bacterium’s defences as R-M systems are not adaptive. Ifthe bacterium is moved to another environment with different bacteriophages or MBEs, the defence system becomes inefficient to retain within the genome. This defence system is not adaptive and is only useful if the invading genetic material has an appropriate restriction site that can be cleaved by the R-M system (Labrie, Samson and Moineau, 2010). The upside to harbouring such a system is that the response time of this defence system is faster and more efficient, which could be a reason these defensive systems are found widely across bacteria (Weissman et al., 2021).

1.7 MEMBRANOME

The membranome plays a crucial role in the protection of the single-celled organism, and the composition of the membranome has been linked with CRISPR-Cas. Research carried out by Rubio et al. (2023) analysed the ESKAPE organism group and found that there was a link between the membranome and CRISPR-Cas. Associations between other cellular systems and CRISPR-Cas are being discovered (Hille and Charpentier, 2016), and this project sought to add to that knowledge base.

The membranome contains multiple types of proteins, including anchored and unanchored proteins (Brown and Waneck, 1992). Anchored proteins are bound to the plasma membrane and are unable to move about the space freely (Brown and Waneck, 1992). These proteins are usually transport proteins, or proteins that have a defensive mechanism to block the binding of external molecules to the plasma membrane. Unanchored proteins fulfil a different role as they are present outside of the inner membrane within the periplasmic space; these proteins are designed to intercept attacks before they reach the inner membrane, and are usually secreted (Fryszczyn et al., 2011) . Other structures outside of the cell-surface membrane are also used as defensive mechanisms to protect from non-specialised attack, these can include antigenic proteins and the peptidoglycan cell wall (Salton and Kim, 2011). The cell wall is designed to protect from environmental factors, whereas the antigenic proteins are designed for signalling what they are and whether they are the same species or not, to other bacteria and organisms within their environment. These structures all work in unison outside of the cell, however, the links found within the research mentioned previously and within this project, may infer that there is more of a link between the defence mechanisms outside of the cell and inside, in this project we attempt to find the associations between the internal defence mechanisms (in this case CRISPR-Cas) and external defensive mechanisms.

1.8 GENOME ANNOTATION AND PANGENOME CREATION

Genome annotation, the identification and characterisation of functional genetic material within a file which contains the genome, is a crucial process in bioinformatics. Annotating genomes can label genes, regulatory elements, and even non-coding sequences; these can be separated into structural annotation and functional annotation (Ejigu and Jung, 2020). Structural annotation labels the genes' locations, coding sequences (CDS) and even CRISPR-Cas arrays. Functional annotation labels sequences with biological roles based off of the homology, experimental and predictive data (Loewenstein et al., 2009). To perform this task, there are applications which allows one to feed genetic information into and annotate quickly and easily; however, these applications require the use of databases to find these sequences. A few databases which are commonly employed for this are: RefSeq and UniProt (Bateman et al., 2020; Goldfarb et al., 2024). The use of sequence similarity searches is also employed to assist in finding predicted labels for genes which are not exactly alike. Blast and Diamond are both frequently used and respected applications used for this task (Altschul et al., 1990; Buchfink, Xie and Huson, 2014). These applications though, have their downsides; incorrect predictions and mis-annotation of genes are both issues which can arise. Another frequent issue is labelling uncharacterised proteins, as there is no information about them, and even in some cases, do not have a name associated with them. Even with these challenges, this process and mentioned applications are pillars in the bioinformatic process for studying genetic variation and improving microbial strain characterisation (Truong et al., 2017).

The use of annotated genomes allows the formation of a pangenome. A pangenome is a collation of all of the genes found within the genomes of many different species; a matrix of presence and absence of each gene family, which separates them into core genes - genes found in most if not all genomes- and accessory genes -genes found in a few genomes (Matthews et al., 2024). To construct the pangenome, the comparison of all of the genomes which are to be added is required, and there are tools available to help with this task: Roary, PanX and PanTA. These applications allow for the construction of pangenomes in effective time windows, and althought the computational requirement is intensive, it can be controlled through the understanding of the size and depth of the pangenome being created. Most applications require a niche type of file to create a pangenome: a .GFF3 file. A GFF3 file contains both the annotated gene sequences and the raw DNA sequence; this type of file can be created from the annotation application being used. Pangenomes are unique in their position within bioinformatics, which allows for the understanding of gene associations and strain-specific genes that may influence virulence or environmental survivability.

1.9 WHY γ-PROTEOBACTERIA?

The focus of this study is on a γ-proteobacteria pangenome. γ-proteobacterial genomic data is readily available, allowing for an exceptionally large dataset to be created for the training of the ML algorithm. Many of these genomic files are available in the required format, and with the appropriate level of contamination and completeness of the genome from the NCBI repository. Reference and complete genomes are the only genomes downloaded from the repository for this project. This study focusses on genomes that have been sequenced after the 1st of January 2018, in order to take advantage of the improved quality of these genomes that is primarily due to the use of next generation sequencing techniques.

The NCBI repository offers a wide range of available genomes for bacterial species and an even spread of the presence of CRISPR-Cas systems within the dataset, which enables the highest predictive power of the algorithm. A dataset with a balanced number of CRISPR-containing and -deficient genomes provides greater power for understanding their association with genes in the respective background genomes.

γ-proteobacteria have been used throughout many different comparative genomic studies. The genetic variability that is shown by this phylogeny lends itself to suitability within large scale comparative genomic datasets (Vázquez-Rosas-Landa et al., 2017). This phylogeny also exhibits high levels of HGT, emphasising the highly variable nature of the genomes (Juhas et al., 2009). These bacteria are becoming more significant within clinical biology and medical sciences due to their relevance within infection cases and the rise in antibiotic resistance (Diebold et al., 2023). γ-proteobacteria are a phylogeny which interacts with many different species of flora and fauna and these microbes are extremely versatile within ecology. Their presence and interactions within certain microbiomes and environments can offer a beneficial outcome for both parties involved (Köberl et al., 2017).

The primary aim for this research project is to develop a RF algorithm in Python, that when provided with a pangenome of γ-proteobacteria, will be capable of finding gene families that have an association with CRISPR-Cas systems. To achieve this aim, a large interspecies pangenome will be constructed using complete genomes from the NCBI repository; this will require the use of PanTA as a pangenome construction tool. This pangenome will be refined and annotated with a binary identifier to determine if a genome does or does not contain CRISPR-Cas arrays. Importantly, the gene families which are already known to have associations with CRISPR-Cas systems will be removed from the pangenome, which will reduce the algorithm's dependency on known associations.

In this thesis, the materials and methods presents a step-by-step overview of our pipeline and the applications used, including all of the settings and data used for each step. A few steps include evidence of the editing of application code, where images and line numbers are referenced regarding the changes that were made. The results section includes: excerpts of each pangenome that illustrate how the dataset for training our algorithm was constructed, the associations found by the algorithm and its final calculated accuracy metrics. In the discussion section of this thesis, you will find the breakdown of the analysis of the algorithm's accuracy, and associations alongside their 'feature importances'. Chi-squared tests were carried out on the contingency tables, shown in our results section, to help understand the significance level of the results gathered.

2

Materials And Methods

All computational work was completed on the Ada high performance computing facilities which are provided by the University of Nottingham (https://exchange.nottingham.ac.uk/blog/introducing-ada-the-universitys-new-most-powerful-hpc-service/) All scripts used and mentioned will be freely available for use and viewing on the GitHub repository (https://github.com/Zephyure/Thesis-Code-1). The high-performance computing system that was used had maximum settings allowed these settings were as follows: 300 Gb of RAM, 96 CPU cores and 2 Tb of hard drive storage. While some scripts which were used did use the full 300 Gb of RAM, most did not. The datasets are available in the supplementary information. Below is the Pipeline overview, which was used to process all the genomes, each step that was used work in succession, however, due to unstandardised file structures some files do not work with this pipeline. For that reason, the starting number of genomes was 14,500, however at the end of the pipeline only 9,689 genomes remained for use in training the ML algorithm. Figure 2.1 shows the entire main application pipeline it does not include all custom scripts which were used due to the number; however, all scripts are available in the GitHub as mentioned. While using applications which require dependencies an environment was used, specifically Anaconda, this allows for installation of the applications and their dependencies in an easy-to-use package where a ‘Conda’ environment can be used to separate different processes and applications depending on the versions of the dependencies which are needed.

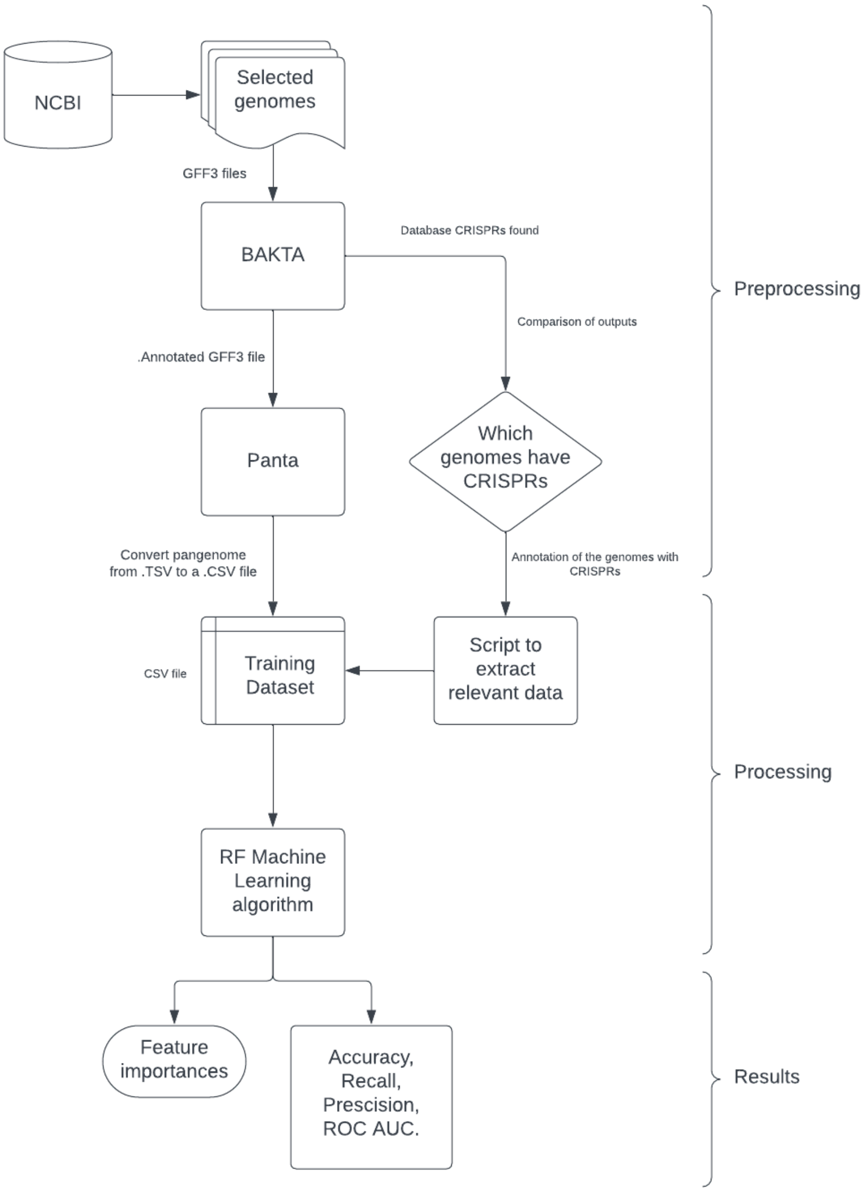
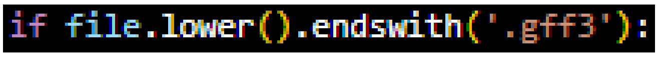

NCBI
Selected genomes
GFF3 files
BAKTA
Database CRISPRs found
Comparison of outputs
.Annotated GFF3 file
Panta
Which genomes have CRISPRs
Convert pangenome from .TSV to a .CSV file
Annotation of the genomes with CRISPRs
CSV file
Training Dataset
Script to extract relevant data
RF Machine Learning algorithm
Feature importances
Accuracy, Recall, Prescision, ROC AUC.
Preprocessing
Processing
Results

Figure 2.1: Pipeline Overview

2.1 GENOME ACQUISITION

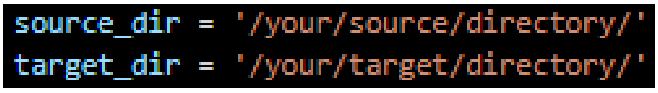
NCBI datasets application was used to interact and query with the NCBI database, using specified criteria to download high quality starting data. The command used: ’datasets download taxonomy taxon 1236 –assembly-source Refseq –assembly-level complete –include gff3 –released-after 2018-01-01’ this command downloaded 14500 gff3 file format genomes through the NCBI database API.

2.1.1 MOVING FILES WITH SPECIFIC FILE ENDINGS



```
if file.lower().endswith('.gff3'):
```

After downloading these genomes, we needed a way to move specific files from the download package to the directory that we wanted to store them in.



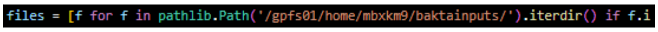
```
source_dir = '/your/source/directory/'
target_dir = '/your/target/directory/'
```

These lines allow for the changing of the source directory and target directory, this script was used at every step of this project to move all the files around simply and quickly. (The FileMover.py script is available here: https://github.com/Zephyure/Thesis-Code-1/blob/main/FileMover.py)

2.2 GENOME ANNOTATION USING BAKTA

Bakta (Schwengers et al., 2021) was used as a genome annotation application to annotate the (14,500) genomes to generate the file formats which were required to be used within the pan-genome generation application PanTA (Le et al., 2024) which requires a .gff3 file format with the genome sequence in fasta format underneath a detailed annotation of the genome. Bakta has a database option which must be downloaded and the file path to the database was integrated into the command to activate bakta. A job-array of 100 simultaneous jobs was used to run all (14500) genomes through bakta quickly. Bakta does not have a parallelisation command within it so each genome must be run separately, for this a custom script written in Python was used to allow for automatic annotation this script can be found in the GitHub repository under the name BaktaPipe2.py. Bakta outputs many files which vary in use cases. We only retained the .json files and .gff3 files for further analysis. BAKTA was installed onto a Conda environment and the ‘Diamond’ dependency required a downgrade from the latest version back to version (2.7) due to a conflict with the more recent version (Buchfink, Xie and Huson, 2014). Ensuring the installed dependencies are compatible with Bakta was required for proper usage of the application.

2.2.1 CREATING LIST OF FILES TO BE PROCESSED



```
files = [f for f in pathlib.Path('/gpfs01/home/mbxkm9/baktainputs/').iterdir() if f.i
```

To facilitate the use of Bakta we first iterated through the directory of bacterial genomes downloaded from the NCBI repository. This line in the script makes a list of all file names, to allow for full processing of the genomes.

2.2.2 RUNNING BAKTA PROCESS

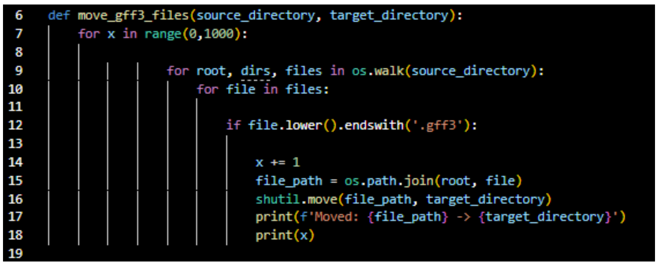
Using the ‘subprocess’ module a Python script was used to run terminal commands with applications that needed to be used through the console. Bakta and most applications for this kind of process require use of this kind of command. This script also renames the output files of Bakta to the name of the input genome, Bakta produces a whole file batch for each file processed which is why we use the file mover script to move the files that we want after full annotation has been carried out. The movement of ‘.gff3’ files into their own directory for the use within pangenome creation, and the ‘.json’ files are moved as well for use in the CRISPR annotation step later. (Available at: https://github.com/Zephyure/Thesis-Code-1/blob/main/Baktapipe.py)

2.3 PANGENOME CONSTRUCTION

PanTA (Le et al., 2024) was used as a pan-genome generation tool to generate a gene presence absence file which simply shows all the genomes and all the genes which are present within each genome. This allows for the use of the gene_presence_absence.Rtab in the next steps of the pipeline. The ‘.GFF3’ files were input into the PanTA application 1000 at a time, this was due to the limitations of the high-performance computing system that was available. PanTA was used is due to its ability to add a new genome or multiple genomes to an already generated pangenome. This functionality allows for the gradual increase of the size of the pan-genome, 1000 genomes each time. The gene_presence_absence.Rtab was then reformatted into a comma-separated values file and was then binarized to show 1 for gene presence and 0 for gene absence.

2.3.1 MOVING 1000 FILES AT A TIME

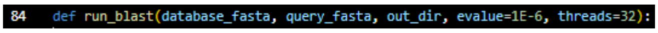
To move 1000 files at a time into the pangenome target directory a for loop with a counter to 1000 was implemented, and a file type filter was used so that it would only move files of a specific file type, in this case ‘.gff3’ files, this enables the directory to be populated with 1000 files at a time from the source directory.



```
def move_gff3_files(source_directory, target_directory):
    for x in range(0,1000):

                for root, dirs, files in os.walk(source_directory):
                    for file in files:

                        if file.lower().endswith('.gff3'):

                            x += 1
                            file_path = os.path.join(root, file)
                            shutil.move(file_path, target_directory)
                            print(f'Moved: {file_path} -> {target_directory}')
                            print(x)

```

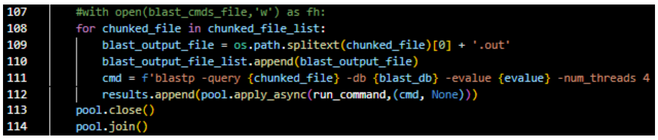
This type of movement uses the ‘Shutil’ module to allow the movement of the files instead of the copying of the files.

2.3.2 PANTA CODE EDITED: (MAIN_PIPELINE.PY)



```
84  def run_blast(database_fasta, query_fasta, out_dir, evalue=1E-6, threads=32):
```

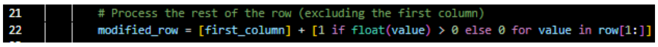
For PanTA to run faster I edited the 84th line of the ‘main_pipeline.py’ to increase the number of ‘threads’ that PanTA uses for BLAST commands to 32, the number of ‘threads’ refers to the number of CPU cores that will be used when the application is being ran. This effects the number of CPU cores which can be pooled by the PanTA application so the number of cores which I needed from the Ada HPC was changed to enable this level of threads.



```
    #with open(blast_cmds_file,'w') as fh:
    for chunked_file in chunked_file_list:
        blast_output_file = os.path.splitext(chunked_file)[0] + '.out'
        blast_output_file_list.append(blast_output_file)
        cmd = f'blastp -query {chunked_file} -db {blast_db} -evalue {evalue} -num_threads 4
        results.append(pool.apply_async(run_command,(cmd, None)))
    pool.close()
    pool.join()
```

Like the previous change however this was to change the ‘num_threads’ value to 4 so that PanTA would run 8 BLASTs at a time for much faster processing of the genomes and genes which are being added to the pangenome. Without these two changes PanTA took too long to sequentially add 1,000 genomes at a time. Upon investigation changing to 4 cores being allotted for each BLAST search within the PanTA application and a 32 core CPU pool allowed for the greatest speed-up while keeping the CPU usage to a minimum.

Even though we do not use ‘Diamond’ within the PanTA application, the program has an option to use it. A compatibility issue with the version of ‘Diamond’ that was being used by the application, required us to downgrade from the latest version of ‘Diamond’ back to a more stable version.



```
        # Process the rest of the row (excluding the first column)
        modified_row = [first_column] + [1 if float(value) > 0 else 0 for value in row[1:]]
```

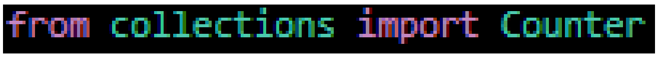
2.3.3 BINARIZING DATASET

After PanTA was used, we had to then binarize the data set to be able to feed it into our algorithm, this required changing any value which was greater than '0' to be changed to a '1'. PanTA has many different values across the pangenome instead of just a presence-absence (or 1 and 0) approach to evaluating genomes, it runs a blast across the genomes and finds the number of genomes which contain a similar gene sequence for that genome‘s version of the gene family and outputs that value as it being present; For instance a value for a gene family within the genomes column of ‘42‘ depicts that 42 genomes have a similar sequence to that genome‘s version of the gene. The absence of the gene family is still only outputted as '0'. This script can be found in the GitHub repository at: https://github.com/Zephyure/Thesis-Code-1/blob/main/GenomeBinarizer.py.

2.4 REMOVAL OF EXTRANEOUS DATA POINTS

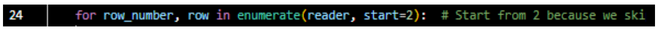
The 2% check script uses the data within the pangenome to check whether there is a minimum of a 2% difference in the number of genomes which contain or do not contain said gene. At least 2% of the columns must contain a different value to the rest of the dataset (within a row) to be kept within the dataset. The script then removes all the rows which do not contain the required difference, this script was designed to remove the cases where only one genome contains a gene, because it would be a result that is not statistically relevant.

2.4.1 COUNTING TO CALCULATE



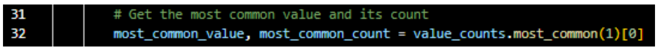
```
from collections import Counter
```

The ‘Counter’ module was used in this scenario to reliably count the number of ‘1’s or ‘0’s found in each row of the pangenome.



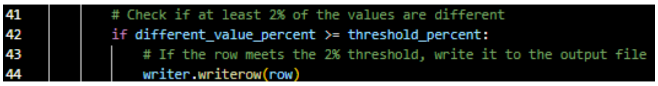
```
    for row_number, row in enumerate(reader, start=2):  # Start from 2 because we ski
```

We skip row 1 and column 1 or the pangenome due to the first row being used for the names of the genomes and column 1 contains the feature labels for each feature (gene family label) for the algorithm.



```
            # Get the most common value and its count
            most_common_value, most_common_count = value_counts.most_common(1)[0]
```

Line 32 finds the most common value of either '1' or '0' within each row, then assigns that value to the most common value, and also assigns the number of times that value was found to the most common count variable to then be used to calculate the percentage which the value covers within that row.

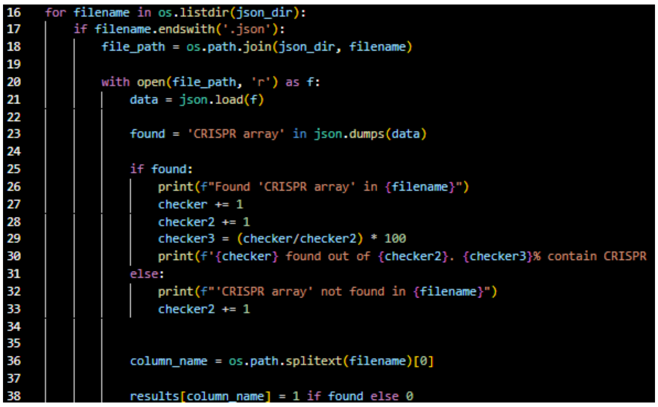


```
41          # Check if at least 2% of the values are different
42          if different_value_percent >= threshold_percent:
43              # If the row meets the 2% threshold, write it to the output file
44              writer.writerow(row)
```

Line 42 calculates if the coverage of the row is more than 2% of the different values, which then if it is found to be greater than or equal to the threshold set it then uses line 44 to write the row into a new '.csv' file which is the filtered pangenome. This threshold can be changed to any value which you desire but for this project it was set to 2% allowing extraneous data points to be removed to give the algorithm more reliable predictive capabilities. This script is available at: https://github.com/Zephyure/Thesis-Code-1/blob/main/twopercentcheck.py.

2.5 ANNOTATING PANGENOME WITH CRISPR DATA POINTS

This script was written in Python to use the .json files that were generated by the Bakta application to annotate each of the genomes within the `gene_presence_absence.CSV` based off of whether the string 'CRISPR array' was found within the .json file which has the same name as the column, it then places a 1 for if the desired string was found or a 0 if it was not found in a new row at the bottom of the .CSV file.

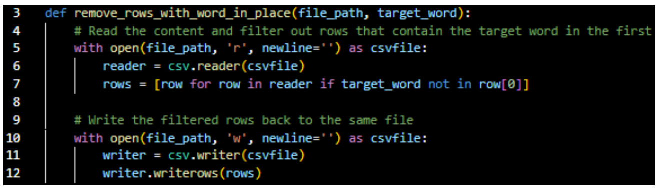


```
16  for filename in os.listdir(json_dir):
17      if filename.endswith('.json'):
18          file_path = os.path.join(json_dir, filename)
19
20          with open(file_path, 'r') as f:
21              data = json.load(f)
22
23              found = 'CRISPR array' in json.dumps(data)
24
25              if found:
26                  print(f"Found 'CRISPR array' in {filename}")
27                  checker += 1
28                  checker2 += 1
29                  checker3 = (checker/checker2) * 100
30                  print(f'{checker} found out of {checker2}. {checker3}% contain CRISPR
31              else:
32                  print(f"'CRISPR array' not found in {filename}")
33                  checker2 += 1
34
35
36              column_name = os.path.splitext(filename)[0]
37
38              results[column_name] = 1 if found else 0
```

This is a part of the script which was used to annotate the pangenome with CRISPR classifications under each genome, this if what the algorithm will use to learn how to identify if CRISPR will be present or not. From this script we can see that in line 16 we iterate through the directory with the '.json' files from Bakta which we moved earlier. Line 21 loads the file data and allows us to use line 23 to search through to find the words ‘CRISPR array’ this indicates where Bakta has found a CRISPR array which in this project we say that CRISPR is present in the genome if this is found within the ‘.json’ file. From there we then made a running count from lines 27-29 where the exact percentage of genomes which contain CRISPR was calculated in real time. This script then populated the CRISPR row within the pangenome with either a 1 or 0 based on if the ‘CRISPR array’ quote was found within its associated ‘.json’ file or not. This script is available here: https://github.com/Zephyure/Thesis-Code-1/blob/main/Crispradder.py.

2.6 REMOVING CRISPR ASSOCIATED GENES

A Python script written to remove any gene from the pangenome that contained ’Cas’ within its nomenclature. This would remove genes such as the ’cas2’ gene and its representative row within the .csv file. Some ’CRISPR associated proteins’ genes were left within the pangenome specifically any ’csy’ gene such as ’csy3’ these genes were left in the pangenome to set a benchmark to check if the algorithm was finding similar associations each time the algorithm was ran.



```python
def remove_rows_with_word_in_place(file_path, target_word):
    # Read the content and filter out rows that contain the target word in the first
    with open(file_path, 'r', newline='') as csvfile:
        reader = csv.reader(csvfile)
        rows = [row for row in reader if target_word not in row[0]]

    # Write the filtered rows back to the same file
    with open(file_path, 'w', newline='') as csvfile:
        writer = csv.writer(csvfile)
        writer.writerows(rows)
```

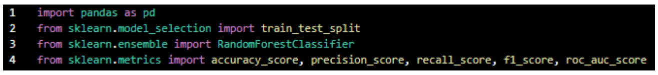
The script used in this step is available for use from the GitHub repository: https://github.com/Zephyure/Thesis-Code-1/blob/main/CasRemover.py.

2.7 RANDOM FOREST ALGORITHM

The SKLearn random forest package was used to write the ML algorithm, in conjunction with the .CSV file that was constructed previously. Transposition of the data was required for compatibility with the SKLearn package. The random forest package from SKLearn can produce many different statistics, however, this algorithm was written to only save certain statistics. For the results of this project the feature importance of all genes within the pangenome were collected into a .CSV file and the calculated Accuracy, Precision, Recall, F1 score and ROC AUC scores in another .CSV file.

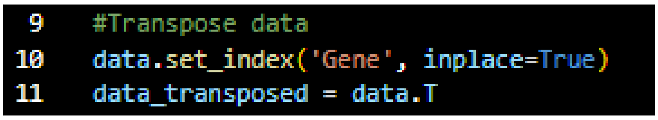
2.7.1 CODE USED FOR RANDOM FOREST ALGORITHM

The ’pandas’ module allows for the creation of a ’pandas’ data frame to allow for the data to be manipulated in ways which it needs to be used for the algorithm



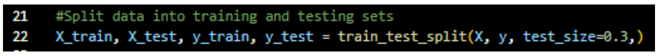
```
1    import pandas as pd
2    from sklearn.model_selection import train_test_split
3    from sklearn.ensemble import RandomForestClassifier
4    from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score
```

to work, this option does have an issue though, if the data frame is too large, say the pre 2% removal pangenome with too many cells in the matrix 'pandas' will be unable to create and manipulate the data frame. The 'sklearn' module was used to simplify the RF algorithm to allow for simple creation and control of each individual setting if required. Line 3 is where the 'RandomForestClassifier' was imported.



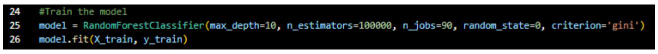
```
9    #Transpose data
10   data.set_index('Gene', inplace=True)
11   data_transposed = data.T
```

Here we take the data frame and set an index; this index will be used to make sure that the 'features' (the gene families within the pangenome) are still labelled when the data frame is transposed from vertical to horizontal, in line 10 we see 'inplace=True' this setting states that the label is already in the correct place on the Un-transposed data frame and will be kept in the same place when transposed, due to the way the random forest algorithm is coded to work in the 'sklearn' library.



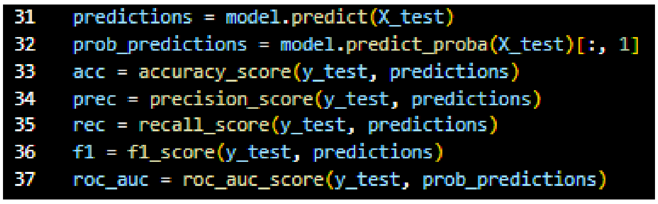
```
21   #Split data into training and testing sets
22   X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,)
```

The data was split into a training set and testing set. In line 22 the 'test_size' is set to 0.3 meaning that 30% of the data is set aside in the training of the algorithm so that it can be used to test the algorithm after training, this can be changed to any specific number from 0-1, however, it was found in testing that a split of 0.2 (20%) to 0.3 (30%) was best for the algorithm's accuracy.



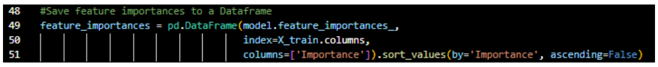
```
24   #Train the model
25   model = RandomForestClassifier(max_depth=10, n_estimators=100000, n_jobs=90, random_state=0, criterion='gini')
26   model.fit(X_train, y_train)
```

These lines of code run the algorithm on our dataset, this sets the settings for the algorithm as well, 'max_depth' is the depth that each decision tree will grow to until moving to the next one. 'n_estimators' is the direct control over the number of decision trees will be generated and calculated by the algorithm, the depth and number of trees are a direct link to how many CPU cores you will need as well as the amount of RAM you will require. Consequently, we set the 'n_jobs' to 90 to use 90 CPU cores, this was used in conjunction with 300 Gb of RAM which was available to the algorithm. The criterion for classification was set to 'gini' which refers to the Gini Index this was chosen due to testing carried out where the best performance of the algorithm was found when ’gini’ was used as the criterion.



```
31  predictions = model.predict(X_test)
32  prob_predictions = model.predict_proba(X_test)[:, 1]
33  acc = accuracy_score(y_test, predictions)
34  prec = precision_score(y_test, predictions)
35  rec = recall_score(y_test, predictions)
36  f1 = f1_score(y_test, predictions)
37  roc_auc = roc_auc_score(y_test, prob_predictions)
```

Lines 32-38 calculate all the metrics which are used to analyse the capabilities of the algorithm, these are then saved into a ‘.csv’ file. These metrics are calculated by the ‘sklearn’ library previously mentioned, using this package allows us to calculate the metrics directly from the algorithm.



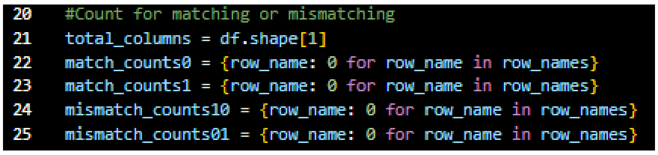
```
#Save feature importances to a Dataframe
feature_importances = pd.DataFrame(model.feature_importances_,
                                   index=X_train.columns,
                                   columns=['Importance']).sort_values(by='Importance', ascending=False)
```

The use of a ’Pandas’ data frame is used again to collect the feature importance values which are calculated from the model, this section places all the calculated values and places them into a descending order based off the calculated ’Importance’ value. This script is available on the GitHub at: https://github.com/Zephyure/Thesis-Code-1/blob/main/RandomForestAlgo.py.

2.8 CREATING CONTINGENCY TABLES FROM THE PANGENOME

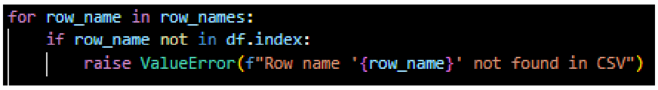
To calculate the Contingency tables for each gene, a Python script was used to find the number of genes for all associations between any gene and the ‘CRISPR’ identifier row.

2.8.1 CODE USED TO CONSTRUCT CONTINGENCY TABLES



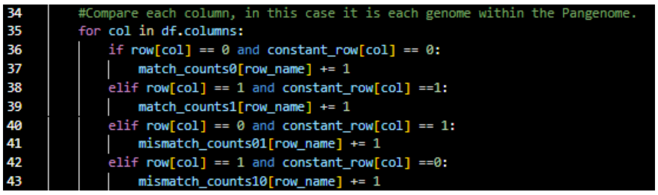
```
20  #Count for matching or mismatching
21  total_columns = df.shape[1]
22  match_counts0 = {row_name: 0 for row_name in row_names}
23  match_counts1 = {row_name: 0 for row_name in row_names}
24  mismatch_counts10 = {row_name: 0 for row_name in row_names}
25  mismatch_counts01 = {row_name: 0 for row_name in row_names}
```

Line 21 is the creation of a dataframe which enables the searches, and lines 22-25 create arrays for each option which the contingency table has. The variables named ‘match_counts0’ and ‘match_counts1’ both are related to the times where both the gene being searched for and the ‘CRISPR’ identifier row have the same value within the genome. Whereras, the variables ‘mismatch_counts10’ and ‘mismatch_counts01’ link to when the gene is present and ‘CRISPR’ is absent, and when the gene is absent but ‘CRISPR’ is present respectively.



```
for row_name in row_names:
    if row_name not in df.index:
        raise ValueError(f"Row name '{row_name}' not found in CSV")
```

This checks if the genes that are being compared to ‘CRISPR’ identifier row are actually within the pangenome, this step requires the full name of the gene which is being added to the contingency table.



```
34      #Compare each column, in this case it is each genome within the Pangenome.
35      for col in df.columns:
36          if row[col] == 0 and constant_row[col] == 0:
37              match_counts0[row_name] += 1
38          elif row[col] == 1 and constant_row[col] ==1:
39              match_counts1[row_name] += 1
40          elif row[col] == 0 and constant_row[col] == 1:
41              mismatch_counts01[row_name] += 1
42          elif row[col] == 1 and constant_row[col] ==0:
43              mismatch_counts10[row_name] += 1
```

Lines 35-43 add the counts of each option of the contingency table to their respective variables. This script is available on GitHub at: https://github.com/Zephyure/Thesis-Code-1/blob/main/ContingencyTableMaker.py

3 Results

3.1 PREPROCESSING

The use of NCBI's Datasets allowed for the download of 14,500 genome files from the NCBI public API. These files were used in conjunction with the Bakta application to annotate the genome files using the Bakta associated database (available at (put in links)). The Bakta application outputs nine different file types. Seven of these files were deleted, however, two of these file types were not as they were used for the later steps in the data processing. The results of the annotation and files which were used are available in the supplementary information section. PanTA used the input of .GFF3 files to produce a pangenome, and excerpt of this pangenome can be found in Table 3.1. This output is coupled with Table 3.2 which is the percentage of each type of gene these genes are placed into four categories 'Core genes' which are genes found in 99.0-100.0% of the genomes, 'Soft core genes' which are found within 95.0-98.99 of the genomes, 'Shell genes' which are found within 15.0-94.99% of the genomes and finally the 'Cloud genes' which are found to be in 0-14.99% of genomes. These numbers which are calculated by PanTA allow us to understand how closely related our genomes are within the pangenome that was constructed. There are additional steps between each stage which are within the Materials and Methods section. These steps utilised custom scripts to move file types and sort the respective files into groups for use in further analysis within the pipeline.

3.1.1 REMOVAL OF GENES NOT FOUND IN MORE THAN 2% OF GENOMES

Using the 2% check algorithm on the large pangenome, the removal and trimming of 1,372,990 gene families from the pangenome which left only 28,683 gene families for the algorithm to use to find associations for prediction. This was the expected result from the 2% check, due to the substantial number of genomes which were used to construct the pangenome and the fact that the vast majority of genes in pangenomes tend to be rare (Horesh et al., 2021).

Table 3.1: Excerpt of Pangenome.
A small section of the Pangenome created by PanTA after the input of .GFF3 files, the Gene column shows which gene has been found within each of the genomes. The rows show how many gene sequences were found to be 70% similar to each of the other sequences found for each gene within the pangenome. The full pangenome is available in the electronic appendix.

Gene	GCF_012052965.1	GCF_012053145.1	GCF_012053325.1	GCF_012053725.1
pilA	0	0	0	0
pmgR	0	0	0	0
mazZ	0	0	0	0
stfR	3	2	3	2
ssb	1	1	2	1

3.1.2 EXCERPT OF FIRST PANGENOME

After annotation, 9,689 genomes were added to the pangenome above using the PanTA program. From this, the result which was produced was expected; the genes are organised in descending order of most common to the least common. Each genome has its own column, and each gene family has its own row; many gene families have different variants which are listed within the pangenome as the name of the gene family with a series of numbers afterwards, this is shown more in Table 3.6.

Table 3.2: Pangenome Gene Category Statistics.
The number of genes that fit into each category according to the output from PanTA, before removal of 2% outliers.

Category of Genes	Number of Genes	Percentage of Genes
Core	33	0.002%
Soft Core	21	0.001%
Shell	5328	0.380%
Cloud	1396291	99.6%
Total	1401673	100%

3.1.3 PANGENOME GENE CATEGORY STATISTICS

The Pangenome showed a large distribution of types of genes across the four calculated categories: Core, Soft Core, Shell, and Cloud (Table 3.2). The 33 core genes found are highly conserved across all genomes meaning they were present in 99% of genomes, this was expected within such a diverse dataset. There are fewer Soft-Core genes (21) compared to Core genes. These genes were found in most, but not all of the genomes present in the dataset. The Shell genes (5,328) were found in most genomes, but not all. This is expected when compared to the other categories because it shows a moderate level of genetic diversity. Many of the gene families fell into the Cloud genes category. This distribution of genes fits the expectations from previous studies, where the majority of genes are in the cloud category, and also because of the large amount of genetic variability present in the group of genomes that we added to the pangenome. Large bacterial pangenomes show this level of separation due to the highly variable nature of these genomes.

Table 3.3: Excerpt of Pangenome After Binarization.

Gene	GCF_012052965.1	GCF_012053145.1	GCF_012053325.1	GCF_012053725.1
pilA	0	0	0	0
pmgR	0	0	0	0
mazZ	0	0	0	0
stfR	1	1	1	1
ssb	1	1	1	1

3.1.4 EXCERPT OF PANGENOME AFTER BINARIZATION

After the initial pangenome was constructed, a binarization step was carried out by the Python script 'genomebinarizer.py'. This transformed the dataset into the presence absence matrix (Table 3.3) which was the structure required for input into the algorithm. This process converted all the data within the pangenome into either a '1', indicating the presence of the gene, or a '0' which indicated absence. An example of this is the gene 'stfR' which across the 4 genome columns, shows presence (1), in contrast, the genes 'pilA', 'pmgR' and 'mazZ' show only absence '0' in all genome columns. This excerpt is merely 4 columns of the 9689 within the pangenome. Across the rest of the pangenome the gene families found here are the most conserved gene families which aligned with the expectation. This binarization process enabled further analysis with the use of the RF algorithm.

Table 3.4: Excerpt of the Pangenome After CRISPR Identification.
The ellipses indicate a skip in the information as the list of genes is 17813 genes long. CRISPR row has been added to the bottom using the CRISPRadder.py algorithm.

Gene	GCF_012052965.1	GCF_012053145.1	GCF_012053325.1	GCF_012053725.1
pilA	0	0	0	0
pmgR	0	0	0	0
mazZ	0	0	0	0
stfR	1	1	1	1
ssb	1	1	1	1
...	...	...	...	...
CRISPR	1	1	1	1

3.1.5 CRISPR ANALYSIS OF PANGENOME

The ‘CRISPRadder.py’ Python script was used to process the binarized pangenome to add the ‘CRISPR’ row. This followed the same logic of presence ‘1’ and absence ‘0’. This type of addition to the pangenome enabled the use of our RF algorithm, as every genome which was annotated in the first step of pre-processing had labelled sequences within a ‘.json’ file. This file contains the labelled ‘CRISPR array’ sequences. This step also allowed for the understanding of the percentage of the pangenome that contained CRISPR, and did not contain CRISPR. The number of instances of CRISPR within the genomes was 4,914 (50.72%) present, with 4,775 (49.28%) genomes recording CRISPR loci being absent. This split is near perfect for the algorithm, due to the desire of having a 50-50 weighting of present and absent genomes.

Table 3.5: Calculated Accuracy, Precision, Recall, F1 Score and AUC-ROC
The calculated values of the set measurements of accuracy for the Random forest algorithm

Accuracy	Precision	Recall	F1 Score	AUC-ROC
0.89	0.93	0.85	0.89	0.96

3.1.6 CALCULATED ACCURACY METRICS

The algorithm ran on the processed pangenome, which enabled the evaluation by use of classification metrics: Accuracy, Precision, Recall, F1 score and AUC-ROC (Table 3.5). The model achieved an Accuracy score of 0.89, indicating that 89% of all predictions were correct. This score shows the model has strong accuracy. The Precision score of 0.93, indicates that when the model predicted that a genome would contain CRISPR, it was correct 93% of the time. The Recall value of 0.85 is measured based off the proportion of true positive results (genomes which contain CRISPR) that were classified correctly. This small differential between precision and recall imply that the model will prioritise classifying genomes as negative to avoid false positives whilst still having strong sensitivity. The calculated F1 score is the mean of precision and recall. The F1 score of 0.89, showed that the model was capable. The final metric calculated was the AUC-ROC score, 0.96, this score highlights the model as having excellent capabilities when distinguishing between genomes which contain CRISPR and do not contain CRISPR. The AUC-ROC score near 1.0, implies the model is good at discriminating between the classes of positive and negative. The results show promise of an algorithm which can classify with great reliability, high accuracy, and strong precision.

Table 3.6: Feature Importance Table.
The calculated feature importance of each gene within the top 10 highest feature importances, however, the entire entire pangenome has been calculated for based off of the genes used within the decision trees in the random forest.

Gene	Feature importance
pbp4b	0.00286
csy3	0.00276
yghA	0.00256
ydeI_17612	0.00251
pdeA	0.00226
lsrA_07895	0.00219
csy3_11537	0.00216
ais	0.00203
ycaM	0.00197
yjfL_11048	0.00197

3.1.7 ALGORITHM CALCULATED FEATURE IMPORTANCES

The RF algorithm was used to calculate the feature importance of genes when predicting the presence of CRISPR-Cas systems in the genomes. Feature importance values are determined by the algorithm and calculated based on the frequency of each gene when contributing to the decision splits within each decision tree. The genes which have the highest feature importance contribute most to the algorithm's predictive capabilities. Table 3.6 contains the list of the top 10 genes found by the algorithm; pbp4b has the highest feature importance value of 0.00286, followed by the positive control csy3 (0.00276), yghA (0.00256), and ydel_17612 (0.00251). According to the algorithm, these genes were found to reduce the impurity in classification decisions, playing a significant role in the classification of genomes. These values are small, however, across the algorithm using these genes in multiple decision trees supplements the predictive capabilities of the model.

The presence of csy3 gene variants (both csy3 and csy3_11537) imply that the algorithm is finding genes which associate with CRISPR-Cas systems. The csy3 gene and its variants remained in the pangenome as positive controls, guaranteeing that while testing the algorithm, the identified genes were found alongside known associates of CRISPR. These results confirm that the algorithm can find genes that contribute to the accuracy of the model more than other genes. This means that they have a stronger association with CRISPR-Cas systems than others.

Table 3.7: Contingency tables and chi-squared tests for the top 10 genes removing the positive controls

Gene Name	Gene present	CRISPR present	Gene + CRISPR present	Neither present	Chi-squared statistic	Degrees of freedom	p-value
pbp4b	93	2482	2432	4682	966.091	1	<0.01
yghA	336	2113	2801	4439	521.056	1	<0.01
ydel_17612	94	2441	2473	4681	2907.894	1	<0.01
pdeA	222	2270	2644	4553	2809.268	1	<0.01
lsrA_07895	107	2585	2329	4668	2623.565	1	<0.01
ais	321	2212	2702	4454	2627.949	1	<0.01
ycaM	232	2410	2504	4543	2539.568	1	<0.01
yjfL_11048	337	2222	2692	4438	2566.845	1	<0.01

3.1.8 SIGNIFICANCE FOR TOP 10 GENES WITHIN THE PANGENOME

The calculated chi-squared statistics for all genes is found to be relatively high, indicating a large difference when compared to the expected distributions. All p-values are found to be <0.01 enabling the rejection of the null hypothesis for all genes, the p-values were too small to be calculated, implying great significance for the association between the genes and CRISPR. 'ydel_17612' and pdeA have the highest chi-squared statistics, these genes have the furthest deviation from the expected distributions. All of the genes except 'pbp4b' and 'yghA' have a chi-squared statistic which is very significant (>2500) these results greatly support the hypothesis of significant associations found by the algorithm. 'pbp4b' has a lower chi-squared statistic (966.091), however, this result implies that the association is still significant. The lowest chi-squared statistic (521.056) is found for 'yghA', but it is still a result which points to a statistically significant association. This table also removes the positive control variables, these are found by the algorithm but are known to have a significant association with CRISPR.

Table 3.8: Contingency tables and chi-squared tests for Penicillin binding protein families

Gene Name	Gene present	CRISPR present	Gene + CRISPR present	Neither present	Chi-squared statistic	Degrees of freedom	p-value
pbp4b	93	2482	2432	4682	966.091	1	<0.01
pbpG	2452	978	3936	2323	889.68	1	<0.01
pbpG_01091	475	4874	40	4300	399.62	1	<0.01
pbpC	351	4873	41	4424	263.22	1	<0.01
pbpC_15607	153	4789	125	4622	3.30	1	0.026
pbpC_01805	1865	1606	3308	2910	776.08	1	<0.01
pbpG_07529	465	4798	116	4310	232.53	1	<0.01
pbpC_03747	245	4822	92	4530	75.63	1	<0.01
pbpC_50525	209	4624	290	4566	11.21	1	<0.01

3.1.9 SIGNIFICANCE OF 'PBP' HOMOLOGOUS FAMILY GENES WITHIN THE PANGENOME

Table 3.8 contains the results of chi-squared tests carried out on the association between the Penicillin-Binding Protein (PBP) gene family and CRISPR presence-absence throughout the bacterial pangenome. The contingency tables include the observed counts of genomes which contain and do not contain PBP genes and the CRISPR presence-absence. There is a range of chi-squared results across the PBP gene family, from 3.79 for pbpC_15607 to 889.68 for pbpG, only one result from these statistical tests imply that the results are insignificant. The p-value for pbpC_15607 is 0.026 implying that the correlation between this gene and CRISPR presence or absence is not significant, however, the other p-values indict a strongly significant association with CRISPR presence or absence. These p-values for genes other than pbpC_15607 allow us to reject the null hypothesis and represent these genes not being independent from CRISPR, implying a pattern of co-evolution and functional associations between them.

4 Discussion

Using PanTA as a pangenome generation application allowed for the generation of the largest dataset of this kind to be created involving (number of genomes) across the γ-proteobacterial clade. The combination of the large dataset and the appropriate methods of analysis allows the interpretation of the associations between different genes across all environmental and genetic factors. When generating the pangenome from the genome files provided, PanTA does something slightly different when populating its presence absence file to other pangenome creation tools and applications. It populates the cells where the gene has been found with the number of gene sequences showing significant similarity to the sequence found within that specific genome. For example, the genome GCF_012052965.1 contained a sequence for the gene stfR which manifested 95% sequence similarity to three sequences for stfR found in the other (number of all genomes). Binarizing the pangenome provided ensures the information is in the appropriate format for subsequent analyses. Reformatting the data was carried out using a custom script to change all cells within the .CSV file that are populated with values that are larger than 0, changing them to a 1 instead.

The gene type breakdown allows for the understanding of the percentage of all the genes found within the pangenome which fall into the categories defined by the PanTA application settings. A total of 32 core genes were removed, as they were universally core. This is in line with the expectation of a set of approximately 33 ribosomal proteins that are present in all organisms (Melnikov, Manakongtreecheep and Söll, 2018). Due to the enormous size of the dataset, a 2% minimum cut off was required to limit and reduce the number of gene families present in the pangenome. This is because using the cohort of rare gene families that are present or absent in less than 2% of the population of the pangenome would only hinder the capabilities of the algorithm. Rare genes would not show strong statistical associations in any direction. After pruning the rare genes from the dataset, the number of gene families found within the pangenome dropped dramatically to 28,651.

CRISPR identification was added to the final row of the pangenome which was required for the use of the algorithm; however, these values were added based off of the .json files which were produced from the initial annotation of the bacterial genomes. The .json files contain all the genetic sequences which are found by Bakta which could be of use from an annotation point of view, this includes the key word ’crispr array’ which is how the script used the saved .json to add the CRISPR identified row to the pangenome.

The analysis of the pangenome using the random forest algorithm resulted in a general accuracy value of 0.89 to be calculated. Although there is no explicit test for significance in this kind of analysis, the result can be interpreted as an 89% accuracy value and it allows the interpretation of the associations that have been found by the algorithm. Many of the genes with high levels of association with CRISPR have not been previously defined in the datasets as CRISPR-associated genes. The more interesting result from Table 3.5 is the calculated AUC-ROC score. This value implies that the algorithm has excellent discrimination when classifying the binary data. This value also suggests to us that the algorithm has remarkably high likelihood of being able to predict whether a CRISPR locus is in the genome, simply by analysing the rest of the genome. As a rule of thumb within the ML space an AUC-ROC score of 0.8 is good and a score of above 0.9 is seen as excellent, however, the results are dependent on the situation that the algorithm is being used to classify.

When looking at the values of the Precision and Recall, both are lower than the AUC-ROC score implying that there is progress to be made with regards to classification accuracy overall. Improving Precision and Recall could be achieved by including more genomes to increase diversity within the pangenome. Spending more time testing and tuning hyperparameters, or even introducing a different feature selection process to add weightings to the most impactful genes. Alternatively, using different ML strategies could offer improvement, especially using deep learning technologies.

From the results so far, we can say that, as a binary classification algorithm it performs at a high standard, however, as previously stated, the AUC-ROC score being above 0.9 may not be a solid indicator of what the algorithm can achieve with ideal hyper-parameters and datasets. Analysing where the algorithm is failing regarding false positives and false negatives could provide an insight into the issues the model is having trouble with.

Nonetheless, this result suggests that this approach is usable and could be a step towards a system that predicts gene interaction across a clade or even across bacterial life. Incorporating phylogeny and protein-protein interactions could offer a more accurate algorithm, however, this would require standardisation of information across bioinformatics. A multi-class classification system could be explored, to increase the predictive power from only binary classifications to sub-type classification. Identifying groups of genes which interact with CRISPR-Cas systems would allow weightings to be added to the algorithm, increasing accuracy and predictive power.

In Table 3.6 the calculated feature importance of each gene family within the pangenome is shown; these values allow us to understand how the gene families are being used to predict the presence of CRISPR. Within the pangenome, the genes known to be CRISPR associated genes have been removed, except for csy3 and other ’csy’ genes, these were left as positive controls. The research carried out by Zhang et al. (2020) characterized csy3 as a ’Cas’ protein specifically cas7f which is found within the csy protein complex. The fact that the algorithm found this protein to be associated with CRISPR suggests that the algorithm is finding connections that are not only CRISPR associated proteins, but proteins which are found within many different biosynthetic pathways.

Many genes in the top 100 genes regarding feature importance are found to be "uncharacterised", suggesting that these genes have not been characterised in research. However, some are ’uncharacterised’ but also have a function which has been identified using the structure of the protein which is a product of transcribing the genes. As an example of this in Table 3.6 we find ’yghA’ which can be found in the Uniprot repository as an ’uncharacterised oxidoreductase’ (Bateman et al., 2020). Understanding uncharacterised genes is crucial to the development of further research on this subject; however, there are other genes that were found by the algorithm which do have a characterised function. The breakdown of these are found below.

4.1 ANALYSIS OF ’PBP4B’

pbp4b is a gene within the penicillin binding protein family. This gene encodes a DD-carboxypeptidase protein, which was previously believed to be a protein crucial for cell growth and cell division (Vega and Ayala, 2006). However, Vega and Ayala (2006) demonstrated that when this gene and other DD-carboxypeptidases are eliminated, cell growth and cell division of enterobacteria are not negatively affected. An overview of penicillin binding proteins carried out by Sauvage et al. (2008) likens the structure of pbp4b to the structure of ampH, however, there is no current research that characterises a function for the protein encoded in pbp4b. In contrast, ampH has been found to be associated with peptidoglycan recycling. The similarities in the encoded protein structure indicate a potential function of pbp4b; if pbp4b has a similar role to AmpH within the cell, such that it recycles peptidoglycan molecules which are found in the bacterial cell wall, pbp4b may be a part of the membranome. Although there is currently no research to support this claim.

The Penicillin binding protein family is vast and varied, with many different proteins all serving a different role within the cell. Within the pangenome used for our algorithm there are 9 gene families that are labelled as being members of the homologous ’pbp’ family, for these a contingency table and chi-squared test (chi-squared contingency table) was carried out. The results of these statistical tests are found in Table 3.7. Particularly interesting results show that two different sequences of pbpC, specifically the ones labelled pbpC and pbpC_1805; These two sequences of the same gene have opposing results within their respective contingency tables. The contingency table for pbpC shows a large push towards the gene and CRISPR having an avoidance relationship. The gene being present alone in 351 genomes, and both the gene and CRISPR being present in 41 genomes; This is a stark contrast to pbpC_01805, where the gene is present alone in 1865 genomes, but both the gene and CRISPR are present in 3308 genomes. This implies there is more of a correlative nature to the relationship between the variant sequence of pbpC, pbpC_01805, and CRISPR comparatively with the sequence of the main variant of pbpC. This relationship could also be caused by the level of representation of both genes within the genome pool.

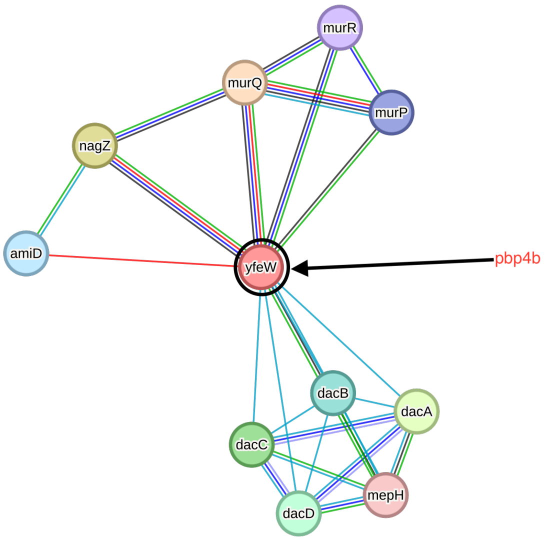


Figure 4.1: StringDB gene network of pbp4b
The light blue and pink lines both represent known interactions between genes. Green, red and dark blue coloured lines indicate connections through gene neighbourhood, gene fusions and gene co-occurrence respectively. Black indicates a connection through co-expression.

Above is the gene network analysis of pbp4b. This network indicates significant associations with genes that are involved in peptidoglycan synthesis and cell wall restructuring (Figure 4.1). The gene of interest, pbp4b, encodes Penicillin Binding Protein 4 (PBP4); this gene network shows many strong interactions with the dacB gene. The gene "dacB" encodes D-alanyl-D-alanine carboxypeptidase, an enzyme which is involved in catalysing reactions that involve DD-carboxypeptidase and DD-endopeptidase. This enzyme plays a critical role in recycling peptidoglycans (Bateman et al., 2020). Furthermore, pbp4b is linked with mrcB, which encodes PBP1B, another member of the penicillin binding protein family. PBP1B is a bifunctional enzyme that exhibits both transglycosylase and transpeptidase activity which assist in the polymerisation of peptidoglycans (King et al., 2017). The network creation also highlighted both mrdA and mrdB, which encode PBP2 and RodA respectively. Both proteins are found to assist in maintaining cell wall integrity and rod shape during elongation. These interactions suggest that pbp4b may play a significant role in predicting the presence of CRISPR in the genome, due to its association with external defence mechanisms.

4.2 OTHER GENE ANALYSIS

'pdeA' is another top 10 gene which encodes a putative c-di-GMP phosphodiesterase, which has been characterised as a part of the inner membrane proteome (Bateman et al., 2020)(Daley et al., 2005). This protein interacts with the cyclic di-DMP signalling molecule which interacts with a large amount of intercellular and extracellular functions, some interesting ones are:

4.2.1 THE GENE 'AIS'

The gene 'ais' is found within the top 10 of the algorithm's feature importance table (Table 3.6) this gene encodes a Lipopolysaccharide core heptose (II)- phosphate phosphatase (Bateman et al., 2020) this protein is found to be within the periplasmic space of gram-negative bacteria therefore making it a part of the membranome of the bacteria. These Lipopolysaccharides are crucial for the survival of gram-negative bacteria, there may be a link here between these proteins and CRISPR-systems. The research carried out by Rubio et al. (2023) found an association between the membranome of bacterial species within the 'ESKAPE' group and CRISPR-Cas systems. These associations could be due to the nature of what CRISPR defends from within the bacteria, an assault through genetic material being injected into the cell. The first layer of defence in this scenario would be the membrane, a link found between CRISPR and membrane-bound proteins would be expected, these proteins having an association with CRISPR opens a a research gap that requires deeper exploration. Could CRISPR be integral for bacterial organisms which have these kinds of lipopolysaccharides but are missing other accessory genes?

This network is for the gene ais which encodes a lipopolysaccharide core heptose(II)-phosphate phosphatase; if it is present this gene may be a member of

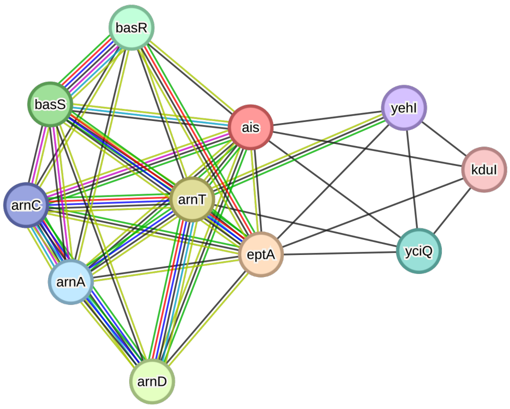


Figure 4.2: StringDB gene network of ais
The light blue and pink lines both represent known interactions between genes. Green, red and dark blue coloured lines indicate connections through gene neighbourhood, gene fusions and gene co-occurrence respectively. Black indicates a connection through co-expression.

the outer membrane of bacterial organisms (Bateman et al., 2020). Specifically, the encoded protein dephosphorylates heptose(II), is a part of lipopolysaccharide metabolism. Within this network we see associations with the arn gene family, these genes (arnT, arnC, arnA and arnD) all take part in modification of lipid A through the attachment of L-Ara4N. These proteins work within the periplasmic space (Lin et al., 2014). Although these genes are all found on the same operon, itis particularly interesting that the other three genes (arnB, arnE and arnF) are not found within this gene network (Lin et al., 2014). Other genes found here are the basS and basR genes, which encode a 2-protein regulatory system involved in the organism's ability to sense environmental stimulus for the organism (Liu et al., 2022).

4.2.2 THE GENE 'LSRA'

lsrA, an intruding gene, has many functions at the intracellular and extracellular level. It is part of the lsr operon, a cluster of genes regulated by the signaling molecule AI-2. Among these, *lsrA* encodes a component of the AI-2 transporter (Xavier and Bassler, 2005). This transporter is found to be within the membrane mosaic, suggesting more genes that this algorithm is finding are parts of the membranome. While this lsrA is a secondary sequence to the main sequence of lsrA, it carries the same responsibilities. lsrA is found to be linked with ATP hydrolysis and ATP binding activity, as the transportation of AI-2 is active and therefore requires ATP to function(Xavier and Bassler, 2005). The research carried out by Xavier and Bassler in 2005 also shows more links from the gene lsrA to the ABC transporter complex, the ATP-binding cassette (ABC) transporter is found within the membranome as well.

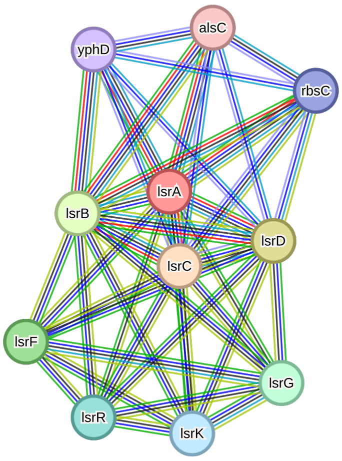


Figure 4.3: StringDB gene network of lsrA.
The light blue and pink lines both represent known interactions between genes. Green, red and dark blue coloured lines indicate connections through gene neighbourhood, gene fusions and gene co-occurrence respectively. Black indicates a connection through co-expression.

The lsrA gene encodes a component of the Lsr (LuxS-regulated) transporter, which is part of an ATP-binding cassette (ABC) transporter complex responsible for importing autoinducer-2 (AI-2), a key molecule in quorum sensing (Xavier and Bassler, 2005). This transport system also includes other components encoded by lsrB, lsrC, and lsrD, all of which work together to facilitate the uptake of AI-2. The gene lsrR enrcodes a repressor protein that is involved in the regulation of the lsr operon. While AI-2 is absent, lsrR binds to the lsr operon promoter region halting transcription (Xue et al., 2009). The lsr coalition of proteins bind and package AI-2 within the periplasmic space, to be transported into the cell for further use. Other than the lsr operon, other genes are found within the network. The gene alsC encodes the permease component of the AlsABC transporter system, which transports D-allose, an uncommon sugar. The rbsC gene encodes another permease protein that is part of the RbsABC transporter complex, involved in transporting D-ribose; this gene would enable the use of ribose as an energy source for the organism if the gene is present.

The fact that the results have found genes which are members of the membranome, reinforces research mentioned earlier. Rubio et al. (2023) infers there is a connection here that should be researched further. Our research and research carried out by Rubio et al. (2023) found correlations between the membranome and CRISPR-Cas systems, although laboratory-based work would be required to find specific connections between them. Some genes homologous to those identified by the algorithm, show differing patterns in their correlation with CRISPR presence or absence within the pangenome. As shown in Table 3.8, certain penicillin-binding proteins appear to be consistently absent when CRISPR systems are present, suggesting a potential avoidance pattern. Particularly looking at both ’pbpG_01091’ and ’pbpC’, these genes show a strong proclivity to avoid CRISPR. However, looking at the variant sequence of ’pbpC_01805’ we can see the opposite; an association can be seen from the results of the pangenome. The gene ’pbpC_01805’ was not identified as significant by the algorithm, and thus cannot be used as a predictor of CRISPR. After carrying out a Chi-squared test on the contingency tables shown in Table 3.8, all results were deemed significant except for one: ’pbpC_15607’. We can assume that this was insignificant as there was not an adequate number of genomes where the gene was present, or where the gene was present in addition to CRISPR. The gene was found in only 278 genomes, representing just 2.8% of the entire pangenome. To better understand the relationship between this gene sequence (pbpC_15607) and CRISPR, more genomes containing pbpC_15607 would be needed. It’s low frequency may simply reflect under representation within the current dataset.

5 Conclusion

The project aimed to understand and analyse the associations between genes and CRISPR-Cas systems in bacterial genomes, using a RF algorithm to find patterns within a bacterial pangenome with 9689 genomes within it. This study found that there are associations to be found between certain genes which some were uncharacterised but other genes namely: pbp4b, ais and lsrA were characterised but had no known association with CRISPR-Cas systems. Using this research, we attempted to understand further why the proteins which are encoded by these genes would interact with CRISPR-Cas systems and found that many the top genes were found to exist in the periplasmic space of bacteria, the area between the outer membrane and the inner membrane. This link indicated that CRISPR-Cas systems may require certain external protection systems to enable the level of defence that the bacteria require from the systems.

Understanding the associations CRISPR-Cas systems have using this method allow for a large amount of data to be analysed and interpreted, this project found that there are associations to be researched further. Understanding CRISPR-Cas systems enables the understanding of the defensive arsenal that bacteria use to survive in the environments they are capable of living in, defending themselves from natural bacteriophages and MGEs. This study also enables the understanding in clinical settings of whether a bacteria will have CRISPR-Cas systems which could be a limiting factor when searching for therapeutic responses to AMR bacterial strains in human infections, this application shows a methodology which could enable a rapid response to the types of infection which are currently believed to be immune to conventional treatment options.

The Feature importance results (Table 3.6) reinforced knowledge from research carried out by Rubio et al. (2023) which linked CRISPR-Cas systems with the membranome of bacteria including certain genes which encode proteins dwelling in the periplasmic space. These results also imply that there are strong associations with external proteins, understanding the connections with proteins which are linked to maintaining the peptidoglycan cell wall imply that the environmental defensive mechanisms also play a role in the selection of CRISPR-Cas systems as an adaptive defence mechanism. Certain genes within the top 10 genes that the algorithm found were unfortunately uncharacterised proteins, these proteins would be interesting to research and eventually characterise with the systems.

This study provides deeper insight into the associations with CRISPR-Cas systems and the other genes found within bacterial genomes, however, there are some limitations with the methodology. These limitations cause the types of CRISPR-Cas system to not be separated and analysed alone, this is due to the lack of time for analysis of types and the data available for each type of CRISPR would be difficult to equalise. The requirements of data for the algorithm to limit over-fitting of types and their associations, the amount of extra preprocessing required would also require standardised file types and file structures. The pipeline worked to analyse the majority of the genomes, however, there were some issues found with applications and how the file structures effected the way that annotation of those genomes happened; the file structures caused these particular genomes to not work with the annotation application (Bakta) causing a loss of 5000 genomes from the pangenome generation step. There are also limitations found in the associations which were searched for, adding analysis of untranslated regions, promoters and enhancers could offer a deeper understanding of why certain genes associated with CRISPR-Cas systems. The phylogeny of bacteria chosen also causes an issue if you are attempting to predict a different type of bacteria other than (γ-proteobacteria), the associations could be vastly different or even opposing the associations found in this type of bacteria.

Finally, the project demonstrated an algorithm which could predict if CRISPR-Cas systems would be present within a genome, and that there are more associated genes than were previously believed that interact with CRISPR-Cas systems. This project has shown that a ML algorithm could be the way to go when attempting to predict complex systems, paving a way for the bioinformaticians to transform the genetics and genomics world.

Appendix

Electronic Appendix link: 10.5281/zenodo.14988084

Bibliography

1. Abedon, S.T., 2012. Bacterial ‘immunity’ against bacteriophages. *Bacteriophage*, 2(1), pp.50–54. doi:10.4161/bact.18609.

2. Agrotis, A. and Ketteler, R. (2015). A new age in functional genomics using CRISPR/Cas9 in arrayed library screening. *Frontiers in Genetics*, 6. doi:10.3389/fgene.2015.00300.

3. Ahmed, A.A.Q. and McKay, T.J.M., 2024. Environmental and ecological importance of bacterial extracellular vesicles (BEVs). *Science of The Total Environment*, 907, p.168098. doi:10.1016/j.scitotenv.2023.168098.

4. Altschul, S.F., Gish, W., Miller, W., Myers, E.W. and Lipman, D.J., 1990. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3), pp.403–410. doi:10.1016/S0022-2836(05)80360-2. .

5. Bateman, A., Martin, M.-J., Orchard, S., Magrane, M., Agivetova, R., Ahmad, S., Alpi, E., Bowler-Barnett, E.H., Britto, R., Bursteinas, B., Bye-A-Jee, H., Coetzee, R., Cukura, A., Da Silva, A., Denny, P., Dogan, T., Ebenezer, T., Fan, J., Castro, L.G. and Garmiri, P., 2020. UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1), pp.D480–D489. doi:10.1093/nar/gkaa1100.

6. Bava, R., Castagna, F., Lupia, C., Poerio, G., Liguori, G., Lombardi, R., Naturale, M.D., Mercuri, C., Bulotta, R.M., Britti, D. and Palma, E., 2024. Antimicrobial resistance in livestock: a serious threat to public health. *Antibiotics*, 13(6), p.551. doi:10.3390/antibiotics13060551.

7. Beavogui, A., Lacroix, A., Wiart, N., Poulain, J., Delmont, T.O., Paoli, L., Wincker, P. and Oliveira, P.H., 2024. The defensome of complex bacterial communities. *Nature Communications*, 15(1), p.2146. doi:10.1038/s41467-024-46489-0.

8. Boeckaerts, D., Stock, M., Ferriol-González, C., Oteo-Iglesias, J., Sanjuán, R., Domingo-Calap, P., De Baets, B. and Briers, Y., 2024. Prediction of *Klebsiella* phage-host specificity at the strain level. *Nature Communications*, 15(1). doi:10.1038/s41467-024-48675-6.

9. Breiman, L., 2001. Random forests. *Machine Learning*, 45(1), pp.5–32. doi:10.1023/a:1010933404324.

10. Brown, D. and Waneck, G.L., 1992. Glycosyl-phosphatidylinositol-anchored membrane proteins. *Journal of the American Society of Nephrology*, 3(4), pp.895–906. doi:10.1681/asn.v34895.

11. Buchfink, B., Xie, C. and Huson, D.H., 2014. Fast and sensitive protein alignment using DIAMOND. *Nature Methods*, 12(1), pp.59–60. doi:10.1038/nmeth.3176.

12. Camara-Wilpert, S., Mayo-Muñoz, D., Russel, J., Fagerlund, R.D., Madsen, J.S., Fineran, P.C., Sørensen, S.J. and Pinilla-Redondo, R., 2023. Bacteriophages suppress CRISPR–Cas immunity using RNA-based anti-CRISPRs. *Nature*, pp.1–7. doi:10.1038/s41586-023-06612-5.

13. Chen, C. and Liaw, A., n.d. Using random forest to learn imbalanced data. [online] Available at: https://statistics.berkeley.edu/sites/default/files/tech-reports/666.pdf [Accessed 5 Dec 2024]

14. Çorbacıoğlu, Ş.K. and Aksel, G., 2023. Receiver operating characteristic curve analysis in diagnostic accuracy studies: a guide to interpreting the area under the curve value. *Turkish Journal of Emergency Medicine*, 23(4), pp.195–198. doi:10.4103/tjem.tjem_182_23.

15. Cumby, N., Edwards, A.M., Davidson, A.R. and Maxwell, K.L., 2012. The bacteriophage HK97 gp15 moron element encodes a novel superinfection exclusion protein. *Journal of Bacteriology*, 194(18), pp.5012–5019. doi:10.1128/JB.00843-12.

16. Curtis, C., Shah, S.P., Chin, S.-F., Turashvili, G., Rueda, O.M., Dunning, M.J., Speed, D., Lynch, A.G., Samarajiwa, S., Yuan, Y., Gräf, S., Ha, G., Haffari, G., Bashashati, A., Russell, R., McKinney, S., Langerød, A., Green, A., Provenzano, E. and Wishart, G., 2012. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403), pp.346–352. doi:10.1038/nature10983.

17. Diebold, P.J., Rhee, M.W., Shi, Q., Trung, N.V., Umrani, F., Ahmed, S., Kulkarni, V., Deshpande, P., Alexander, M., Thi Hoa, N., Christakis, N.A., Iqbal, N.T., Ali, S.A., Mathad, J.S. and Brito, I.L., 2023. Clinically relevant antibiotic resistance genes are linked to a limited set of taxa within gut microbiome worldwide. *Nature Communications*, 14(1), p.7366. doi:10.1038/s41467-023-42998-6.

18. Dilucca, M., Cimini, G. and Giansanti, A., 2021. Bacterial protein interaction networks: connectivity is ruled by gene conservation, essentiality and function. *Current Genomics*, 22(2), pp.111–121. doi:10.2174/1389202922666210219110831.

19. Ejigu, G.F. and Jung, J., 2020. Review on the computational genome annotation of sequences obtained by next-generation sequencing. *Biology*, 9(9), p.295. doi:10.3390/biology9090295.

20. Fraser, H.B., Hirsh, A.E., Wall, D.P. and Eisen, M.B., 2004. Coevolution of gene expression among interacting proteins. *Proceedings of the National Academy of Sciences*, 101(24), pp.9033–9038. doi:10.1073/pnas.0402591101.

21. Fryszczyn, B.G., Brown, N.G., Huang, W., Balderas, M.A. and Palzkill, T. (2011). Use of periplasmic target protein capture for phage display engineering of tight-binding protein–protein interactions. *Protein Engineering, Design and Selection*, 24(11), pp.819–828. doi:10.1093/protein/gzr043.

22. Goldfarb, T., Kodali, V., Pujar, S., Brover, V., Robbertse, B., Farrell, C., Oh, D.-H., Astashyn, A., Ermolaeva, O., Haddad, D., Hlavina, W., Hoffman, J., Jackson, J., Joardar, V.S., Kristensen, D., Masterson, P., McGarvey, K., McVeigh, R., Mozes, E. and Murphy, M., 2024. NCBI RefSeq: reference sequence standards through 25 years of curation and annotation. *Nucleic Acids Research.* doi:10.1093/nar/gkae1038.

23. Goldstone, J.V., Hamdoun, A., Cole, B.J., Howard-Ashby, M., Nebert, D.W., Scally, M., Dean, M., Epel, D., Hahn, M.E. and Stegeman, J.J., 2006. *The chemical defensome: Environmental sensing and response genes in the Strongylocentrotus purpuratus genome.* Developmental Biology, [online] 300(1), pp.366–384. doi:10.1016/j.ydbio.2006.08.066.

24. Gophna, U., Kristensen, D.M., Wolf, Y.I., Popa, O., Drevet, C. and Koonin, E.V., 2015. No evidence of inhibition of horizontal gene transfer by CRISPR–Cas on evolutionary timescales. *The ISME Journal*, **9**(9), pp.2021–2027. doi:10.1038/ismej.2015.20.

25. Greene, C.S., Tan, J., Ung, M., Moore, J.H. and Cheng, C., 2014. Big data bioinformatics. *Journal of Cellular Physiology*, 229(12), pp.1896–1900. doi:10.1002/jcp.24662.

26. Greene, C.S., Tan, J., Ung, M., Moore, J.H. and Cheng, C., 2015. Erratum: Big data bioinformatics. *Journal of Cellular Physiology*, 231(1), p.257. doi:10.1002/jcp.25077.

27. Gupta, A., Kumar, S. and Kumar, A., 2023. Big data in bioinformatics and computational biology: basic insights. *Methods in Molecular Biology*, pp.153–166. doi:10.1007/978-1-0716-3461-5_9.

28. Gupta, P.K. and Gandhi, M., 2023. Bioremediation of organic pollutants in soil–water system: a review. *Biotech*, 12(2), p.36. doi:10.3390/biotech12020036.

29. Hille, F. and Charpentier, E., 2016. CRISPR-Cas: biology, mechanisms and relevance. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 371(1707), p.20150496. doi:10.1098/rstb.2015.0496.

30. Horemans, S., Pitoulias, M., Holland, A., Soultanas, P. and Janniere, L., 2020. Glycolytic pyruvate kinase moonlighting activities in DNA replication initiation and elongation. *arXiv preprint*, arXiv:2012.06222. doi:10.48550/arxiv.2012.06222.

31. Horesh, G., Taylor-Brown, A., McGimpsey, S., Lassalle, F., Corander, J., Heinz, E. and Thomson, N.R., 2021. Different evolutionary trends form the twilight zone of the bacterial pan-genome. *Microbial Genomics*, **7**(9). doi:10.1099/mgen.0.000670.

32. Huberts, D.H.E.W. and van der Klei, I.J., 2010. Moonlighting proteins: an intriguing mode of multitasking. *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research*, 1803(4), pp.520–525. doi:10.1016/j.bbamcr.2010.01.022.

33. Ishino, Y., Shinagawa, H., Makino, K., Amemura, M. and Nakata, A., 1987. Nucleotide sequence of the *iap* gene, responsible for alkaline phosphatase isozyme conversion in *Escherichia coli*, and identification of the gene product. *Journal of Bacteriology*, 169(12), pp.5429–5433. doi:10.1128/jb.169.12.5429-5433.1987.

34. Jansen, R., van Embden, J.D.A., Gaastra, W. and Schouls, L.M., 2002. Identification of genes that are associated with DNA repeats in prokaryotes. *Molecular Microbiology*, 43(6), pp.1565–1575. doi:10.1046/j.1365-2958.2002.02839.x.

35. Jiang, W., Maniv, I., Arain, F., Wang, Y., Levin, B.R. and Marraffini, L.A., 2013. Dealing with the evolutionary downside of CRISPR immunity: bacteria and beneficial plasmids. *PLoS Genetics*, 9(9), p.e1003844. doi:10.1371/journal.pgen.1003844.

36. Jinek, M., Chylinski, K., Fonfara, I., Hauer, M., Doudna, J.A. and Charpentier, E., 2012. A programmable dual-RNA-guided DNA endonuclease in adaptive bacterial immunity. *Science*, 337(6096), pp.816–821. doi:10.1126/science.1225829.

37. Juhas, M., van der Meer, J.R., Gaillard, M., Harding, R.M., Hood, D.W. and Crook, D.W., 2009. Genomic islands: tools of bacterial horizontal gene transfer and evolution. *FEMS Microbiology Reviews*, 33(2), pp.376–393. doi:10.1111/j.1574-6976.2008.00136.x.

38. Kim, H.S., Kweon, J. and Kim, Y., 2024. Recent advances in CRISPR-based functional genomics for the study of disease-associated genetic variants. *Experimental & Molecular Medicine*, pp.1–9. doi:10.1038/s12276-024-01212-3.

39. King, D.T., Wasney, G.A., Nosella, M., Fong, A. and Strynadka, N.C.J., 2017. Structural insights into inhibition of *Escherichia coli* penicillin-binding protein 1B. *Journal of Biological Chemistry*, 292(3), pp.979–993. doi:10.1074/jbc.M116.718403.

40. Köberl, M., Dita, M., Martinuz, A., Staver, C. and Berg, G., 2017. Members of *Gammaproteobacteria* as indicator species of healthy banana plants on *Fusarium* wilt-infested fields in Central America. *Scientific Reports*, 7(1). doi:10.1038/srep45318.

41. Koonin, E.V. and Makarova, K.S., 2019. Origins and evolution of CRISPR-Cas systems. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 374(1772), p.20180087. doi:10.1098/rstb.2018.0087.

42. Koskella, B. and Brockhurst, M.A., 2014. Bacteria–phage coevolution as a driver of ecological and evolutionary processes in microbial communities. *FEMS Microbiology Reviews*, 38(5), pp.916–931. doi:10.1111/1574-6976.12072.

43. Labrie, S.J., Samson, J.E. and Moineau, S., 2010. Bacteriophage resistance mechanisms. *Nature Reviews Microbiology*, 8(5), pp.317–327. doi:10.1038/nrmicro2315.

44. Le, D.Q., Nguyen, T.A., Nguyen, S.H., Nguyen, T.T., Nguyen, C.H., Phung, H.T., Ho, T.H., Vo, N.S., Nguyen, T., Nguyen, H.A. and Cao, M.D., 2024. Efficient inference of large prokaryotic pangenomes with PanTA. *Genome Biology*, 25(1). doi:10.1186/s13059-024-03362-z.

45. Leptihn, S. and Loh, B., 2022. Complexity, challenges and costs of implementing phage therapy. *Future Microbiology*, 17(9), pp.643–646. doi:10.2217/fmb-2022-0054.

46. Li, Y., Huang, C., Ding, L., Li, Z., Pan, Y. and Gao, X., 2019. Deep learning in bioinformatics: introduction, application, and perspective in big data era. *arXiv preprint*, arXiv:1903.00342. doi:10.48550/arxiv.1903.00342.

47. Lin, Q.Y., Tsai, Y.-L., Liu, M.-C., Lin, W.-C., Hsueh, P.-R. and Liaw, S.-J., 2014. *Serratia marcescens arn*, a PhoP-regulated locus necessary for polymyxin B resistance. *Antimicrobial Agents and Chemotherapy*, 58(9), pp.5181–5190. doi:10.1128/aac.00013-14.

48. Liu, Y., Wang, Y., Chen, X., Jin, J., Liu, H., Hao, Y., Zhang, H. and Xie, Y., 2022. BasS/BasR two-component system affects the sensitivity of *Escherichia coli* to plantaricin BM-1 by regulating the tricarboxylic acid cycle. *Frontiers in Microbiology*, 13. doi:10.3389/fmicb.2022.874789.

49. Loewenstein, Y., Raimondo, D., Redfern, O.C., Watson, J., Frishman, D., Linial, M., Orengo, C., Thornton, J. and Tramontano, A., 2009. Protein function annotation by homology-based inference. *Genome Biology*, 10(2), p.207. doi:10.1186/gb-2009-10-2-207.

50. Lopatina, A., Tal, N. and Sorek, R., 2020. Abortive infection: bacterial suicide as an antiviral immune strategy. *Annual Review of Virology*, 7(1), pp.371–384. doi:10.1146/annurev-virology-011620-040628.

51. Loureiro, A. and da Silva, G., 2019. *CRISPR-Cas: Converting a bacterial defence mechanism into a state-of-the-art genetic manipulation tool. Antibiotics*, [online] 8(1), p.18. doi:10.3390/antibiotics8010018.

52. Łusiak-Szelachowska, M., Międzybrodzki, R., Drulis-Kawa, Z., Cater, K., Knežević, P., Winogradow, C., Amaro, K., Jończyk-Matysiak, E., Weber-Dąbrowska, B., Rękas, J. and Górski, A., 2022. Bacteriophages and antibiotic interactions in clinical practice: what we have learned so far. *Journal of Biomedical Science*, 29(1). doi:10.1186/s12929-022-00806-1.

53. Marraffini, L.A. and Sontheimer, E.J. (2008). CRISPR Interference Limits Horizontal Gene Transfer in Staphylococci by Targeting DNA. *Science*, [online] 322(5909), pp.1843–1845. doi:10.1126/science.1165771.

54. Makarova, K.S. and Koonin, E.V. (2015). Annotation and Classification of CRISPR-Cas Systems. *Methods in Molecular Biology*, [online] 1311, pp.47–75. doi:10.1007/978-1-4939-2687-9_4.

55. Makarova, K.S., Wolf, Y.I., Iranzo, J., Shmakov, S.A., Alkhnbashi, O.S., Brouns, S.J.J., Charpentier, E., Cheng, D., Haft, D.H., Horvath, P., Moineau, S., Mojica, F.J.M., Scott, D., Shah, S.A., Siksnys, V., Terns, M.P., Venclovas, Č., White, M.F., Yakunin, A.F. and Yan, W. (2019). Evolutionary classification of CRISPR–Cas systems: a burst of class 2 and derived variants. *Nature Reviews Microbiology*, 18(2), pp.67–83. doi:10.1038/s41579-019-0299-x.

56. Matthews, C.A., Watson-Haigh, N.S., Burton, R.A. and Sheppard, A.E., 2024. A gentle introduction to pangenomics. *Briefings in Bioinformatics*, 25(6). doi:10.1093/bib/bbae588.

57. McKenzie, R.E., Keizer, E.M., Vink, J.N.A., van Lopik, J., Büke, F., Kalkman, V., Fleck, C., Tans, S.J. and Brouns, S.J.J., 2022. Single cell variability of CRISPR-Cas interference and adaptation. *Molecular Systems Biology*, 18(4), p.e10680. doi:10.15252/msb.202110680.

58. Melnikov, S., Manakongtreecheep, K. and Söll, D., 2018. Revising the structural diversity of ribosomal proteins across the three domains of life. *Molecular Biology and Evolution*, 35(7), pp.1588–1598. doi:10.1093/molbev/msy021.

59. Millar, K.J., 2025. Electronic appendix – *Kyle Millar 'A random forest approach to understanding CRISPR-Cas associations in bacteria'. Zenodo.* doi:10.5281/zenodo.14988084.

60. Monaco, A., Pantaleo, E., Amoroso, N., Lacalamita, A., Lo Giudice, C., Fonzino, A., Fosso, B., Picardi, E., Tangaro, S., Pesole, G. and Bellotti, R. (2021). A primer on machine learning techniques for genomic applications. *Computational and Structural Biotechnology Journal*, [online] 19, pp.4345–4359. doi:10.1016/j.csbj.2021.07.021.

61. Nada, H., Choi, Y., Kim, S., Jeong, K.S., Meanwell, N.A. and Lee, K., 2024. New insights into protein–protein interaction modulators in drug discovery and therapeutic advance. *Signal Transduction and Targeted Therapy*, 9(1). doi:10.1038/s41392-024-02036-3.

62. Nicolaou, K.C. and Rigol, S., 2017. A brief history of antibiotics and select advances in their synthesis. *The Journal of Antibiotics*, 71(2), pp.153–184. doi:10.1038/ja.2017.62.

63. Nooren, I.M.A., 2003. New EMBO member's review: diversity of protein–protein interactions. *The EMBO Journal*, 22(14), pp.3486–3492 doi:10.1093/emboj/cdg359.

64. Ochiai, H. and Yamamoto, T., 2023. Construction and evaluation of zinc finger nucleases. *Methods in Molecular Biology*, pp.1–25. doi:10.1007/978-1-0716-3016-7_1.

65. Rubio, A., Sprang, M., Garzón, A.A., Moreno-Rodriguez, A., Pachón-Ibáñez, M.E., Pachón, J., Andrade-Navarro, M.A. and Pérez-Pulido, A.J., 2023. Analysis of bacterial pangenomes reduces CRISPR dark matter and reveals strong association between membranome and CRISPR-Cas systems. *Science Advances*, 9(12). doi:10.1126/sciadv.add8911.

66. Russel, J., Pinilla-Redondo, R., Mayo-Muñoz, D., Shah, S.A. and Sørensen, S.J., 2020. CRISPRCasTyper: automated identification, annotation, and classification of CRISPR-Cas loci. *The CRISPR Journal*. doi:10.1089/crispr.2020.0059.

67. Salam, M.A., Al-Amin, M.Y., Salam, M.T., Pawar, J.S., Akhter, N., Rabaan, A.A. and Alqumber, M.A.A., 2023. Antimicrobial resistance: a growing serious threat for global public health. *Healthcare*, 11(13), p.1946. doi:10.3390/healthcare11131946.

68. Salton, M.R.J. and Kim, K.-S., 2011. Structure. In: S. Baron, ed. *Medical Microbiology*. 4th ed. [online] Galveston: University of Texas Medical Branch at Galveston. Available at: https://www.ncbi.nlm.nih.gov/books/NBK8477/ [Accessed 15 Apr 2024].

69. Sawa, T., Moriyama, K. and Kinoshita, M., 2024. Current status of bacteriophage therapy for severe bacterial infections. *Journal of Intensive Care*, 12(1). doi:10.1186/s40560-024-00759-7.

70. Schwengers, O., Jelonek, L., Dieckmann, M.A., Beyvers, S., Blom, J. and Goesmann, A., 2021. Bakta: rapid and standardized annotation of bacterial genomes via alignment-free sequence identification. *Microbial Genomics*, 7(11). doi:10.1099/mgen.0.000685.

71. Singh, P.K. and Kundu, S., 2013. Biosynthesis of gold nanoparticles using bacteria. *Proceedings of the National Academy of Sciences, India Section B: Biological Sciences*, 84(2), pp.331–336. doi:10.1007/s40011-013-0230-6.

72. Spirin, V. and Mirny, L.A., 2003. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences*, 100(21), pp.12123–12128. doi:10.1073/pnas.2032324100.

73. Stanley, S.Y. and Maxwell, K.L., 2018. Phage-encoded anti-CRISPR defenses. *Annual Review of Genetics*, 52(1), pp.445–464. doi:10.1146/annurev-genet-120417-031321.

74. Stephenson, A.A., Raper, A.T. and Suo, Z. (2018). Bidirectional Degradation of DNA Cleavage Products Catalyzed by CRISPR/Cas9. *Journal of the American Chemical Society*, 140(10), pp.3743–3750. doi:10.1021/jacs.7b13050.

75. Truong, D.T., Tett, A., Pasolli, E., Huttenhower, C. and Segata, N., 2017. Microbial strain-level population structure and genetic diversity from metagenomes. *Genome Research*, 27(4), pp.626–638. doi:10.1101/gr.216242.116.

76. Vasu, K. and Nagaraja, V., 2013. Diverse functions of restriction–modification systems in addition to cellular defense. *Microbiology and Molecular Biology Reviews*, 77(1), pp.53–72. doi:10.1128/mmbr.00044-12.

77. Vázquez-Rosas-Landa, M., Ponce-Soto, G.Y., Eguiarte, L.E. and Souza, V., 2017. Comparative genomics of free-living *Gammaproteobacteria*: pathogenesis-related genes or interaction-related genes? *Pathogens and Disease*, 75(5). doi:10.1093/femspd/ftx059.

78. Wang, X. and Leptihn, S., 2024. Defense and anti-defense mechanisms of bacteria and bacteriophages. *Journal of Zhejiang University - Science B.* doi:10.1631/jzus.b2300101.

79. Wang, Y., Fan, H. and Tong, Y. (2023). Unveil the Secret of the Bacteria and Phage Arms Race. *International Journal of Molecular Sciences*, [online] 24(5), p.4363. doi:10.3390/ijms24054363.

80. Wang, Z., Gerstein, M. and Snyder, M. (2009). RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1), pp.57–63. doi:10.1038/nrg2484.

81. Weissman, J.L., Alseth, E.O., Meaden, S., Westra, E.R. and Fuhrman, J.A., 2021. Immune lag is a major cost of prokaryotic adaptive immunity during viral outbreaks. *Proceedings of the Royal Society B: Biological Sciences*, 288(1961). doi:10.1098/rspb.2021.1555.

82. Westermarck, J., Ivaska, J. and Corthals, G.L., 2013. Identification of protein interactions involved in cellular signaling. *Molecular & Cellular Proteomics*, 12(7), pp.1752–1763. doi:10.1074/mcp.R113.027771.

83. Wheatley, R.M. and MacLean, R.C., 2020. CRISPR-Cas systems restrict horizontal gene transfer in *Pseudomonas aeruginosa. The ISME Journal.* doi:10.1038/s41396-020-00860-3.

84. Xavier, K.B. and Bassler, B.L., 2005. Regulation of uptake and processing of the quorum-sensing autoinducer AI-2 in *Escherichia coli. Journal of Bacteriology*, 187(1), pp.238–248. doi:10.1128/jb.187.1.238-248.2005.

85. Xue, T., Zhao, L., Sun, H., Zhou, X. and Sun, B., 2009. LsrR-binding site recognition and regulatory characteristics in *Escherichia coli* AI-2 quorum sensing. *Cell Research*, 19(11), pp.1258–1268. doi:10.1038/cr.2009.91.

86. Yang, S., Yalamanchili, H.K., Li, X., Yao, K.-M., Sham, P.C., Zhang, M.Q. and Wang, J., 2011. Correlated evolution of transcription factors and their binding sites. *Bioinformatics*, 27(21), pp.2972–2978. doi:10.1093/bioinformatics/btr503.

87. Zaayman, M. and Wheatley, R.M. (2022). Fitness costs of CRISPR-Cas systems in bacteria. *Microbiology*, 168(7). doi:10.1099/mic.0.001209.

88. Zhang, H. and McCarty, N., 2016. CRISPR-Cas9 technology and its application in haematological disorders. *British Journal of Haematology*, 175(2), pp.208–225. doi:10.1111/bjh.14297.

89. Zink, I.A., Wimmer, E. and Schleper, C., 2020. Heavily armed ancestors: CRISPR immunity and applications in archaea with a comparative analysis of CRISPR types in Sulfolobales. *Biomolecules*, 10(11), p.1523. doi:10.3390/biom10111523.