Prediction of Host-Pathogen Gene Expression From Dual RNA-seq Data during a Bacterial Infection

by

A thesis submitted to the School of Graduate Studies in partial fulfillment of the requirements for the degree of Master of Science Master of Science

Supervisor: Prof. Lourdes Peña Castillo

Department of Computer Science

Memorial University of Newfoundland

January 2024

St. John's Newfoundland

Abstract

Understanding the mechanisms by which bacteria cause disease, such as apoptosis and inflammatory signals, necessitates a comprehensive knowledge of the genes expressed during infection by both the host and the pathogen. Dual RNA-seq technology enables simultaneous detection of transcripts of the pathogen and host during an infection. In this study, we utilized machine learning to predict the expression levels of genes involved in bacterial infection from their RNA sequence using dual RNA-seq data to obtain gene expression levels. We developed two predictive models: one specifically tailored to the host and the other to the pathogen. Results from these models are promising in terms of macro-average F1-score and macro-average Area Under Receiver Operating Characteristic Curve (AUROC) and demonstrate that machine learning can be applied to dual RNA-seq data to predict gene expression levels during bacterial infection, opening new prospects for future research to build upon these methods and insights.

Acknowledgement

I am immensely thankful to all the people and groups who have supported my research journey. I especially appreciate my supervisor, Dr. Lourdes Peña-Castillo, for her constant support, invaluable advice, and extraordinary mentorship during this project. Her motivation has been crucial in guiding my work. I also thank the School of Graduate Studies at Memorial University of Newfoundland and Labrador for providing financial aid that allowed me to focus on developing this thesis. I am also thankful to the Digital Research Alliance of Canada for allowing me access to their resources to obtain the results necessary to complete my work. I acknowledge the use of ChatGPT (https://chat.openai.com/) and Grammarly (https://grammarly.com/) to improve the academic tone and accuracy of language, including grammatical structures, punctuation and vocabulary of my work. ChatGPT prompts asked it to polish my own text and never to generate text from scratch.

Contents

List of Figures

List of Tables

List of Abbreviations

DOWN Down-regulated 26

M. tuberculosis *Mycobacterium tuberculosis* 62

ND Not differentially expressed 26

S. pyogenes *Streptococcus pyogenes* 64

UP Up-regulated 26

Y. ruckeri *Yersinia ruckeri* 8

List of Acronyms

AUC Area Under Curve 54

AUROC Area Under Receiver Operating Characteristic Curve i

CNN Convolutional Neural Network 10

DEG Differentially Expressed Genes 2

DGE Differential Gene Expression 1

DNA Deoxyribonucleic Acid 4

ENA European Nucleotide Archive 20

FDR False Discovery Rate 8

GEO Gene Expression Omnibus 11

GO Gene Ontology 40

GTEx Genotype-Tissue Expression 11

HPI Hours Post-Infection 16

KNN K-Nearest Neighbors 11

LightGBM Light Gradient Boosted Machine 31

log2FC Log2 Fold Change 24

MAE Mean Absolute Error 11

MOI Multiplicity of Infection 16

mRMR Minimum Redundancy Maximum Relevance 30

NGS Next-Generation Sequencing 9

PCA Principal Component Analysis 30

Chapter 1

Introduction

Eukaryotic cells are susceptible to various types of infections, ranging from viruses and bacteria to eukaryotic parasites like fungi and protozoa [1]. During bacterial infections, a complex interplay occurs between the bacteria and host eukaryotic cells as they navigate their survival and defense strategies [2]. Understanding the interactions between host cells and bacterial pathogens is crucial for advancing therapeutics, diagnostics, and the development of new drugs. One way to unveil these interactions is performing Differential Gene Expression (DGE) analysis to identify host and bacterial genes that show significant changes in expression levels between infected and uninfected cells.

RNA-seq is a technique to get abundance of RNA molecules in a biological sample and widely recognized as a powerful technique for analyzing DGE due to its ability to provide quantitative, comprehensive, and unbiased measurements of gene expression,

thereby facilitating the identification of Differentially Expressed Genes (DEG) [1, 3]. However, a limitation of traditional RNA-seq approaches is their inability to simultaneously analyze the pathogen and host cell without physically separating them [4]. To overcome this constraint, recent advancements known as dual RNA-seq [1] and Path-seq [5] have emerged. These technologies enable the concurrent capture of host and bacterial transcriptomes from infected cells, preserving the intricate host-bacteria interactions within the sample [2]. By maintaining this complex interplay, dual RNA-seq empowers researchers to delve into the dynamic relationship between the host and pathogen, opening up new avenues for comprehensive analysis.

Machine learning algorithms have gained significant traction in biological research, encompassing areas such as biological image analysis [6], cancer studies [7], and protein function prediction [8]. Several studies utilizing single-cell RNA-seq have demonstrated the effectiveness of machine learning and deep learning approaches in identifying DEG that are often missed by traditional RNA-seq data analysis techniques [9, 10, 11, 12]. However, current investigations of dual RNA-seq analysis primarily rely on traditional bioinformatics approaches [13, 14, 15].

In this study, we collected and processed nine published dual RNA-seq datasets to develop and evaluate machine learning models capable of predicting host and bacterial DEG during an infection based on their sequence. We categorized DEG into three classes: up-regulated, down-regulated, and not-differentially expressed. The performance of our models was assessed using the macro-average AUROC metric,

which is the average AUROC across all classes, treating each class equally regardless of their distribution in the dataset. For the host RNAs, our models achieved a mean macro-average AUROC score of $71.06\% \pm 1.82\%$ over a 10-fold cross-validation. Similarly, for bacterial RNAs, the models recorded a score of $66.14\% \pm 0.73\%$. While a random classifier obtained a macro-average AUROC score of 50% for both host and bacterial datasets.

This thesis is structured as follows: Chapter 2 provides the background relevant to our study. Chapter 3 describes our research methods. Chapter 4 presents our findings and interprets these results and discusses their implications. Lastly, Chapter 5 provides a summary of the research, its limitations, and suggestions for future work.

Chapter 2

Background

In this chapter, we first provide a biological context before reviewing bioinformatics dual RNA-seq pipelines used in published studies. These studies employed a range of experimental designs and bioinformatics analysis techniques to explore the dynamic interactions between hosts and bacterial pathogens. The last section of this chapter describes the use of machine learning for predicting DEGs.

2.1 Biological Background

Every living organism's blueprint is encoded within its Deoxyribonucleic Acid (DNA). This molecule, with its iconic double helix structure, contains the genetic instructions necessary for life's development, functioning, and reproduction. As the first step for the cells to use the information stored in the DNA, the information contained in the DNA is copied through a process known as transcription into a variety of Ribonucleic

Acid (RNA) molecules, which then perform various functions or are translated into proteins. While structurally similar to DNA, RNA typically exists in a single-stranded form, contains uracil instead of thymine, and plays multiple roles within the cellular environment [16, 17].

Pathogens have the ability to interact with host cells in various ways to multiply and spread in host cells, but these interactions between hosts and pathogens are quite diverse. Pathogens typically start by establishing themselves within the host, either by sticking to or penetrating the surfaces that line the lungs, gut, bladder, and other parts of the body that are directly exposed to the outside world. Some pathogens, such as viruses and certain bacteria, invade host cells to replicate within them using various methods. Bacteria rely on cell adhesion and processes similar to how cells engulf particles (phagocytic pathways). Once inside, these intracellular pathogens look for a suitable environment where they can multiply. They often manipulate the host cell's traffic and utilize the cell's cytoskeleton for moving around within it. Besides affecting individual host cells, pathogens can also change the behavior of the entire host organism in ways that help them spread to new hosts [17].

The dynamic relationship between host and pathogen leads to changes in gene expression patterns in both the host and the microbe [18]. These interactions demonstrate their strategies for surviving and spreading. As the host tries to eliminate the invading microbe, the microbe employs its own tactics to live on and propagate inside the host. This tug-of-war is central to the evolving relationship between pathogens

and their hosts [17].

2.2 Bioinformatics analysis of dual RNA-seq data

Dual RNA-seq analysis has emerged as a tool to study interactions between host and pathogen during an infection [1]. This technique simultaneously captures the genes activated as a response to the infection in both host and pathogen, providing a comprehensive view of the molecular interaction during infection.

Figure 2.1 shows the workflow of a dual RNA-seq analysis pipeline. The sequencing machine will generate reads given as FASTQ files. A read refer to a short sequence of nucleotides obtained from the sequencing machine. The first step in a dual RNA-seq analysis pipeline is often trimming. Since RNA sequencing can produce reads with artifacts or low-quality bases, especially at their ends, trimming ensures that only high-quality, relevant sequences are retained for further analysis.

Following trimming, the next phase is alignment. The cleaned RNA sequences, or reads, are mapped to a reference genome. This process is for determining the origin of each read, be it from the host or the pathogen. By aligning reads to their respective locations on the reference genome, researchers can discern which genes are being expressed and at what levels.

The final step in this pipeline is read counting. Once the reads are aligned to their respective genes, the number of reads mapped to each gene is counted. This quantification provides a measure of gene expression levels. By comparing read counts

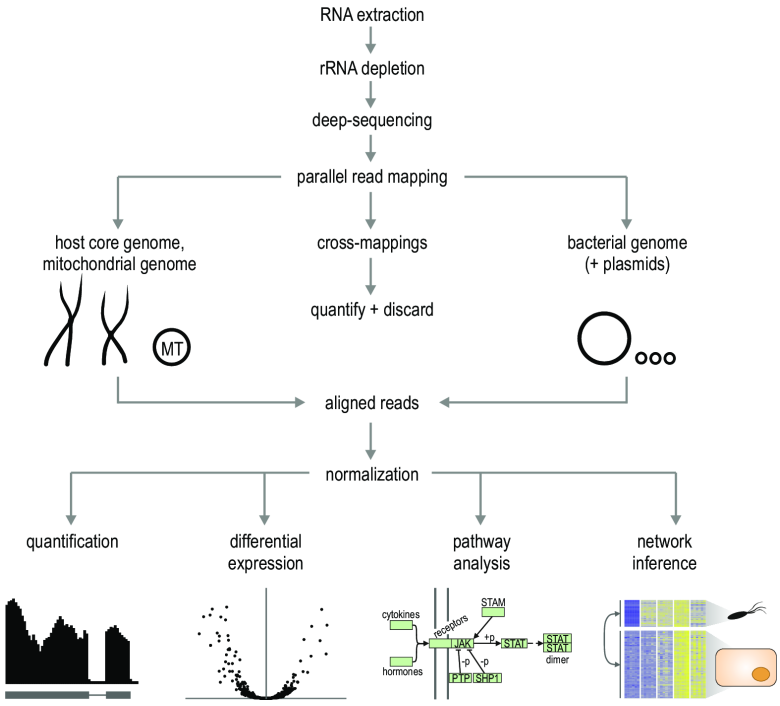

Figure 2.1: Workflow for dual RNA-seq datasets. Figure taken from [14] under CC-BY 4.0.

between different conditions or time points, researchers can identify up-regulated or down-regulated genes, offering insights into the dynamic interplay between host and pathogen during infection.

In several recent studies, various bioinformatics tools and techniques have been utilized. In the initial stage of sequence quality control and trimming, tools such as Trimmomatic [19], FastQC [20], Trim Galore (https://github.com/FelixKrueger/TrimGalore), and cutadapt [21] have been widely used. For instance, the studies by Wu et al. [22], Yang et al. [23], Kachroo et al. [24], Farman et al. [25], and Goldmann et al. [26] employed aforementioned tools to ensure the quality of data for further

analysis.

Moving to sequence alignment, tools like STAR [27], TopHat [28], EDGE-Pro [29], Bowtie2 [30], and segemehl [31] are often selected. Specifically, Wu et al. [22] and Kachroo et al. [24] utilized STAR for mapping reads to the human genome, while Yang et al. [23] used TopHat for aligning reads to the channel catfish (*Ictalurus punctatus*) and *Yersinia ruckeri* (*Y. ruckeri*) genomes. Peterson et al. [5] chose Bowtie2, and Damron et al. [32] used CLC Genomics Workbench (digitalinsights.qiagen.com) which is not commonly done by other studies.

In the gene expression quantification step, various methods are employed across studies. Wu et al. [22] used featureCounts [33], Yang et al. [23] used RSEM [34], and Baddal et al. [35] used HTSeq [36]. A distinct approach was seen in Farman et al. [25] where Salmon [37] was used for quantification.

For DGE analysis, DESeq [38], DESeq2 [39], and edgeR [40] are commonly used across different studies. However, Baddal et al. [35] employed limma [41], showcasing a unique choice among these studies.

The criteria for identifying DEG also varied among different studies. For instance, Wu et al. [22] identified DEG using a False Discovery Rate (FDR) $p-value < 0.05$ and the absolute of the logarithm base 2 of the fold change ($|log2FC| \geq 1$) as criteria, while Yang et al. [23] set a two-fold change and a $p-value < 0.05$ as the threshold for DEG identification. On the other hand, Peterson et al. [5] applied a more stringent cutoff with $|log2FC| > 1$ and multiple hypothesis-adjusted $p-value < 0.01$.

In other miscellaneous steps and tools, Peterson et al. [5] employed a three-stage alignment pipeline using the R package DuffyNGS (https://github.com/robertdouglasmorrison/DuffyNGS), a unique approach in this context. Farman et al. [25] used RUVseq [42] in R for removing unwanted variations, which was not observed in other studies.

Comparative assessments, such as those by K.S.Mehta et al. [43], Li et al. [44] and Schaarschmidt et al. [45], provide critical insights into the performance and suitability of various RNA-seq analysis tools. K.S.Mehta et al. [43] underscores the importance of quality control in Next-Generation Sequencing (NGS) data for accurate disease diagnosis, focusing on the cleaning phase to remove unwanted sequences. Among the tools discussed, FastQC provides a comprehensive quality profile of reads, Trimmomatic excels in trimming and cropping Illumina data, Trim Galore uses the functionality of Cutadapt and FastQC to address specific sequencing datasets, and Cutadapt removes adapters and primers in an error-tolerant manner.

The choice of DESeq, DESeq2, and edgeR for RNA-seq differential analysis can be justified based on their performance in different distribution scenarios according to Li et al. [44]. For RNA-seq count data with a negative binomial distribution, DESeq2 demonstrated slightly better performance than other methods. In the scenario where RNA-seq count data followed a log-normal distribution, both DESeq and DESeq2 were recommended due to better control of the FDR, power, and stability across different conditions. edgeR, on the other hand, did not exhibit strong performance

compared to DESeq2 and DESeq under the same conditions but is still among the commonly utilized methods for differential expression analysis.

Schaarschmidt et al. [45] evaluates seven RNA-Seq alignment tools, revealing that each tool has distinct strengths. STAR and HISAT2, being splice-aware aligners, exhibited high accuracy in mapping reads against a reference genome, with STAR achieving the highest fraction of mapped reads among all tools. On the other hand, Salmon showed a high level of consistency in identifying differentially expressed genes, suggesting its reliability for differential gene expression analysis. The commercial software CLC provided divergent results in DGE analysis, indicating a difference in normalization and statistical tests used [45].

2.3 Machine Learning for Gene Expression Prediction

By now, we have reviewed bioinformatics tools to analyze dual RNA-seq data to identify host and pathogen DEG during infection. Here, we review three studies that used machine learning and deep learning techniques for gene expression prediction.

To identify DEG from RNA-seq data using machine learning, Kakati et al. [46] proposed DEGnext, a Convolutional Neural Network (CNN) based model to predict up-regulated and down-regulated genes using gene expression data sourced from The Cancer Genome Atlas (TCGA) database [47]. Moreover, to overcome the challenge

of small sample sizes and the absence of appropriate labels inherent to RNA-seq data, the authors incorporated transfer learning. This technique leverages patterns learned from related data, making DEGnext adaptable to new datasets without the need for retraining from scratch. When evaluated, DEGnext showcased reliable results, achieving AUROC scores between 88% and 99%, outperforming or matching traditional machine learning methods such as Decision Tree, K-Nearest Neighbors, Random Forest, Support Vector Machine, and eXtreme Gradient Boosting (XGBoost).

Li et al. [48] introduces an algorithm based on XGBoost [49] to predict gene expression values. The dataset used in this paper is a Gene Expression Omnibus (GEO) [50] dataset and a RNA-Seq expression data, which was from the Genotype-Tissue Expression (GTEx) project [51]. When evaluated on the GEO data, the XGBoost algorithm showcasing superior performance, achieving a lower Mean Absolute Error (MAE) than Linear Regression and K-Nearest Neighbors (KNN), on 91.5% of the target genes. Furthermore, regarding overall error on the validation and test sets, XGBoost obtains an error of 0.280 and 0.282, respectively, which was lower than the other models. For instance, KNN obtains errors of 0.586 on the validation set and 0.587 on the test set. In additional testing on RNA-Seq expression data, XGBoost again outperforms the other methods. XGBoost achieves 0.439, while KNN achieves 0.652 in terms of overall error.

Avsec et al. [52] addresses the challenge of predicting gene expression from non-coding DNA sequences using a deep learning architecture named Enformer. This

architecture, inspired by transformers used in natural language processing, can integrate information from long-range interactions in the genome, up to 100kb away. The Enformer model achieved an increase in gene expression prediction accuracy, moving from a correlation of 0.81 to 0.85, edging closer to the experimental-level accuracy of 0.94.

Although these studies successfully utilized machine learning for gene expression prediction, there is a gap in examining gene behavior during infections. Our research fills this gap by examining how host and pathogen genes behave during infections, categorizing them as either up-regulated, down-regulated, or not differentially expressed.

2.4 Summary

In this chapter, we first reviewed the relevant biological background on bacterial infections. We then delved into current practices used to analyze dual RNA-seq data. Lastly, we reviewed how machine learning is used for predicting gene expression levels. To the best of our current knowledge, the prediction of gene expression levels in the course of an infection utilizing RNA sequences annotated according to RNA-seq data remains unexplored. Therefore, the goal of this research is to develop a species-agnostic system that can predict gene expression levels in both the host and bacteria during infection.

Chapter 3

Methods

The primary objective of this study was to predict host and bacterial gene expression levels during a bacterial infection from their RNA sequence. To do that, we labeled the expression level of each host and pathogen gene using dual RNA-seq data and generated a separate model for host and pathogen. The main steps taken are described below:

1. **Data Collection:** We collected ten dual RNA-seq gene expression datasets from a variety of host cells infected by different bacterial species. Data collection is described in Section 3.1.

2. **Data Preprocessing:** The raw RNA-seq data was preprocessed using a custom bioinformatics pipeline. This pipeline involved read quality control and trimming, genome alignment, and read counting. Following this, we performed DGE analysis to identify genes with statistically significant changes in expression be-

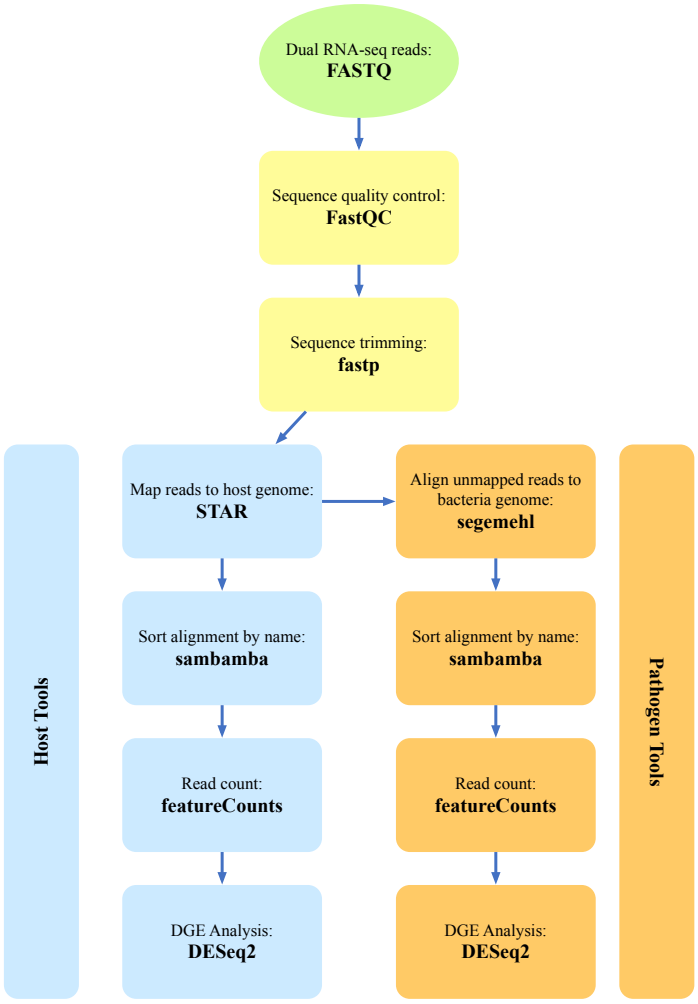

Figure 3.1: Workflow of our dual RNA-seq analysis pipeline.

tween infected and uninfected conditions. Data preprocessing is described in Section 3.2. The visualization of this pipeline is provided in Figure 3.1. Upon finishing the DGE analysis, we performed data labeling and encoded the sequences into numerical features.

3. **Machine Learning Model Training and Selection:** The preprocessed data was used to train several machine learning models for host and bacteria samples

separately. These models were trained using features obtained from numerical encodings of the RNA expression reads. We performed hyper-parameter tuning to determine the optimal settings for each model. The models were then ranked based on their performance, in terms of macro-average AUROC and macro-average F1-score, and the top-performing models were selected for further analysis. Machine Learning model training and selection are described in Section 3.3.

3.1 Data Collection

The data used in this study was collected through a literature search of published dual RNA-seq datasets. Our search strategy involved two main approaches:

1. **Keyword Search:** We conducted a search in PubMed [53] using the following keywords: dual RNA-seq, host-pathogen interaction, and bacterial infection. This search was aimed at identifying studies that performed dual RNA-seq on host cells infected with a bacterial species.

2. **Citation Review:** We also reviewed all papers that cited the work by Westermann et al. [1], which first introduced dual RNA-seq. This allowed us to identify additional studies that may have used dual RNA-seq but did not necessarily include our specific keywords in their title or abstract.

Through these combined search strategies, we identified nine datasets that met the

following criteria:

1. The dataset must include dual RNA-seq data from infected host cells and control samples for host and bacteria.

2. The dataset's metadata must provide sufficient information about the experimental conditions, including the host species, bacterial species, and their genome accessions.

3. The genome file and its corresponding genome annotation for the studied organisms (both host and bacteria) must be available in the NCBI Reference Sequence Database (RefSeq) database [54].

4. The dataset must meet certain quality standards. Specifically, the host mapping rate must be above 50%, indicating a sufficient level of host gene expression data. Additionally, the pathogen mapping rate must not be zero, ensuring that there is detectable bacterial gene expression.

These datasets represented a variety of host cells, bacterial species, culture conditions, and Multiplicity of Infection (MOI). MOI refers to the ratio of infectious agents (here bacteria) to infection targets (here host cells). The specific details of each dataset, including the host species, bacterial species, MOI, Hours Post-Infection (HPI), and other relevant information are provided in Table 3.1.

Table 3.1: Collected dual RNA-seq datasets for our study. Datasets 1, 4, 5, 6, 8 and 9 were selected for training and 2,3 and 7 for validating. HUVECs: Human Umbilical Vein Endothelial, BMDM: Bone-Marrow-Derived Macrophage, NHBE: Normal Human Bronchial Epithelial, HMC-1: Human Mast Cell line 1, THP-1: Tohoku Hospital Pediatrics-1.

No.	Study	Dataset	Pathogen		Host			SRA/ENA Run ID			HPI	MOI
		Accession Number	Name	RefSeq Assembly ID	Name	RefSeq Assembly ID	Tissue/Cell type	Pathogen	Host	Infected		
1	Wu et al. [22]	GSE184085	*Porphyromonas gingivalis*	GCF_000010505.1	*Homo sapiens*	GCF_000001405.40	HUVECs	SRR15886831 SRR15886832 SRR15886833	SRR15886837 SRR15886838 SRR15886839	Host only: SRR15886840 SRR15886841 SRR15886842 Pathogen Only: SRR15886834 SRR15886835 SRR15886836	24	100
2	Yang et al. [23]	PRJNA760961	*Yersinia ruckeri* strain YZ	GCF_017498685.1	*Ictalurus punctatus*	GCF_001660625.2	Trunk Kidney	SRR15827093 SRR15827092 SRR15827086	SRR15827085 SRR15827084 SRR15827083	SRR15827089 SRR15827088 SRR15827087	24	-
3	Kachroo et al. [24]	GSE144100	*Streptococcus pyogenes* strain MGAS2221	GCF_012572265.1	*Macaca fascicularis*	GCF_012559485.2	Quadriceps Skeletal Muscle	SRR10954321 SRR10954322 SRR10954323	SRR10954345 SRR10954350 SRR10954355	SRR10954327 SRR10954332 SRR10954337	24	-
4	Peterson et al. [5]	GSE116357	*Mycobacterium tuberculosis* H37Rv	GCF_000195955.2	*Mus musculus*	GCF_000001635.27	BMDM	SRR7444071 SRR7444072 SRR7444073	SRR7444077 SRR7444078 SRR7444079	SRR7444104 SRR7444105 SRR7444106	24	10
5	Baddal et al. [35]	GSE63900	*Haemophilus influenzae* Fi176, Hi176	GCF_004327565.1	*Homo sapiens*	GCF_000001405.40	NHBE	SRR1714478 SRR1714479 SRR1714480	SRR1714496 SRR1714497 SRR1714498	SRR1714499 SRR1714500 SRR1714501	72	100

No.	Study	Dataset	Pathogen		Host			Sequencing Run ID			HPI	MOI
		Accession Number	Name	RefSeq Assembly ID	Name	RefSeq Assembly ID	Tissue/Cell type	Pathogen	Host	Infected		
6	Farman et al. [25]	PRJEB33395	*Bordetella pertussis* Tohama I	GCF_000195715.1	*Homo sapiens*	GCF_000001405.40	THP-1 Cells	ERR3419003 ERR3419004 ERR3419005	ERR3419018 ERR3419019 ERR3419020	ERR3419042 ERR3419043 ERR3419044	12	50
7	Rienksma et al. [55]	PRJEB6552	*Mycobacterium tuberculosis* variant bovis BCG str. ATCC 35733	GCF_000194035.1	*Homo sapiens*	GCF_000001405.40	THP-1 Cells	ERR560450	ERR560452	ERR560444 ERR560446	24	10
8	Damron et al. [32]	PRJNA343201	*Pseudomonas aeruginosa* PAO1	GCF_000006765.1	*Mus musculus*	GCF_000001635.27	Lung Tissue	SRR4279868 SRR4279869 SRR4279872	SRR4279876 SRR4279877 SRR4279878	Host only: SRR4279870 SRR4279871 SRR4279879 Pathogen only: SRR4279873 SRR4279874 SRR4279875	16	-
9	Goldmann et al. [26]	PRJEB43874	*Staphylococcus aureus* subsp. aureus NCTC 8325	GCF_000013425.1	*Homo sapiens*	GCF_000001405.40	HMC-1 Cells	ERR5530739 ERR5530740 ERR5531330	ERR5530731 ERR5530732 ERR5530733	ERR5530707 ERR5530723 ERR5530726	24	5

3.1.1 Downloading Genomes and Annotations

For our dual RNA-seq data analysis, we required complete genome sequences and annotations for both the host and the pathogen. To acquire these resources, we used the `datasets` command from NCBI Datasets' command line tools (version 1.0) [56]. The command we used to download the genome sequence and annotation for each species was as follows:

```
datasets download genome accession <accession...> \
    --include seq,rna,gtf \
    --filename file_name.zip
```

In this command:

- `accession` specifies Assembly or BioProject accession number.
- `--include` specifies the data files to include are genomic sequence, transcript and annotation in GTF format.
- `--filename` specifies a custom file name for the downloaded data package.

3.1.2 Downloading FASTQ Files

The raw sequencing data for each dataset was downloaded from the Sequence Read Archive (SRA) [57] and European Nucleotide Archive (ENA) [58] using the fasterq-dump tool which is part of SRA-Toolkit (version 2.9.6) [59].

3.2 Data Preprocessing

The raw RNA-seq data from each study was preprocessed using a custom bioinformatics pipeline. In the development of our bioinformatics pipeline, we adapted the approach outlined by Marsh et al. [2]. While we used their pipeline as a foundation, we opted to utilize alternative software tools in read trimming, genome alignment, read counting and DGE analysis to increase the accuracy of read mapping and speed. Each step in our pipeline is described in the following sections:

3.2.1 Quality Control

Prior to any preprocessing steps, we performed an initial quality control check on the raw reads using FastQC (version 2) [20].

3.2.2 Read Trimming

Read trimming is a crucial preprocessing step in the analysis of NGS data, including dual RNA-seq data. During the sequencing process, it is common for the quality of the

sequencing reads to drop off towards the end [43]. This is due to technical limitations of the sequencing technology, and it can result in the inclusion of incorrect bases (nucleotides) in the sequencing reads. Additionally, the sequencing process often involves the use of adapter sequences, which are small pieces of DNA that are added to the ends of the sequencing reads to facilitate the sequencing process. However, these adapter sequences are not part of the original biological sample and need to be removed before the data can be analyzed. Read trimming involves removing these low-quality bases and adapter sequences from the sequencing reads.

We chose to use fastp (version 0.23.1) [60] for read trimming over other tools such as Trimmomatic [19] due to its speed, which was found to be approximately 9 times faster than Trimmomatic [61], and active development, ensuring efficient preprocessing and compatibility with the latest sequencing technologies. We used the default parameters of fastp for read trimming.

3.2.3 Genome Alignment

This step is necessary to determine the genomic origin of the sequencing reads. It involves aligning the sequenced reads to a reference genome.

For the host reads, we used STAR (version 2.7.9a) [27] with default parameters. In a comparative assessment study conducted by Yao et al. [62], RNA-Seq data derived from the *Arabidopsis thaliana* accessions Col-0 and N14 were mapped using seven tools: BWA, CLC, HISAT2, kallisto, RSEM, Salmon, and STAR. As illustrated in

Table 3.2: Percentage of mapped reads for each mapper and sample. Results taken from Yao et al. [62].

Sample	BWA	CLC	HISAT2	kallisto	RSEM	Salmon	STAR
Col-0 %	95.9	96.2	98.9	97.2	96.4	97.9	99.5
N14 %	92.4	95.2	94.9	94.2	93.6	94.6	98.1
Total %	**94.1**	**95.7**	**96.9**	**95.7**	**95.0**	**96.3**	**98.8**

Table 3.2, STAR excelled in read alignment compared to the other tools, by mapping 98.8% reads.

Following the mapping of reads to the host genome, the unaligned reads were then mapped to the bacterial genome. For this, we used segemehl (version 0.3.4) [31]. Segemehl is known for accurately aligning short-reads and showed four times better performance in aligning more reads compared to Bowtie2 in the *Saccharomyces cerevisiae* yeast genome [63]. We used segemehl's default parameters for this step. This two-step alignment process maximized the use of our sequencing data, capturing both host and bacterial gene expression.

3.2.3.1 Building Genome Index

Before the alignment of reads can occur, an index of the reference genome must be built. The genome index is essentially a data structure that allows the alignment software to quickly and efficiently find the positions where the sequencing reads match the reference genome.

We utilized STAR and segemehl to generate genome indices for the host and

bacterial reference genomes, respectively. Specifically, in STAR, we employed the `--runMode genomeGenerate` option to direct the software towards the genome indices generation task. For segemehl, the genome indexing was achieved using the `-x` option.

3.2.4 Read Counting

Following the alignment of reads to the reference genomes, the next step in the RNA-seq analysis pipeline is read counting. This process quantifies the number of reads aligned to each gene in the genome, providing a measure of gene expression levels in the sample.

For this task, we utilized featureCounts (version 2.0.3) [33]. featureCounts assigns each read to a genomic feature (in our case, a gene) based on the read's alignment to the reference genome. While featureCounts demonstrates a level of accuracy in quantification of gene expression comparable to HTSeq [64], it significantly outperforms in speed, being 30 times faster in counting reads [65]. In our analysis, we specifically used the `-t gene` option to ensure that reads were assigned to genes, allowing for a precise quantification of gene expression.

Since segemehl outputs a SAM file, we first needed to convert this to a BAM file, a binary version of the SAM file that is more space-efficient and faster to process. We used the `sambamba-view` tool from sambamba (version 0.8.0) [66] for this conversion.

To adhere to the requirements of featureCounts, the generated host and pathogen BAM files needed to be sorted by name. We utilized `sambamba-sort` tool from

sambamba for this purpose.

Additionally, the strandness of the reads, which is necessary for accurate read counting in featureCounts, was inferred using the `infer_experiment.py` script from RSeQC (version 5.0.1) [67].

3.2.5 DGE Analysis

After read counting, the next step in our dual RNA-seq analysis pipeline is DGE analysis. This process involves comparing the gene expression levels between different conditions or groups to identify genes that are differentially expressed. In the context of our study, we are interested in identifying genes whose expression levels change significantly during a bacterial infection compared to a non-infected condition.

In our study, we performed DGE analysis using DESeq2 (version 1.40.1) [39]. DESeq2 is a tool for analyzing count data from high-throughput sequencing assays such as RNA-seq. It uses a model based on the negative binomial distribution to estimate variance and test for differential expression, which allows it to account for both biological and technical variability in the data. In a comparative study conducted by Li et al. [68], DESeq2 exhibited better results in negative binomial and log-normal distributed data in terms of FDR control, power, and stability across all sample sizes, outperforming other methods, including EdgeR and limma.

Two statistics provided by DESeq2 are the Log2 Fold Change (log2FC) and the adjusted p-value (p-adj).

- **log2FC:** This is a measure of the magnitude of change in gene expression between two conditions. It is calculated as the logarithm to the base 2 of the fold change, which is the ratio of the average expression levels between two conditions. Specifically, the log2FC is determined as log2(A/B), where A and B are the expression levels of a given gene in conditions A and B, respectively. A positive log2FC indicates that a gene is up-regulated (i.e., more highly expressed) in condition A compared to condition B, while a negative log2FC indicates that a gene is down-regulated (i.e., less highly expressed) in condition B compared to condition A.

- **Adjusted P-value (p-adj):** This is a measure of the statistical significance of the observed change in gene expression. It is calculated from the p-value, which is the probability of observing a change in expression at least as extreme as the one measured if there were no real difference in expression between the conditions. The p-value is adjusted for multiple testing to control the false discovery rate, which is the expected proportion of false positives among all genes declared differentially expressed. A lower p-adj indicates a higher level of statistical significance.

3.2.6 Data Labeling

For the machine learning part of our study, we labeled our data based on the DGE results. We used less strict thresholds for labeling genes than those usually used in

transcriptional profiling studies to increase the number of labeled genes for training. Furthermore, we later filtered those genes for consistency, so we needed a larger pool of genes to have enough data to train our machine learning classifier. We classified genes as Up-regulated (UP) if they had a p-adj less than 0.1 and a log2FC greater than or equal to 1. Genes with a p-adj less than 0.1 and a log2FC less than or equal to -1 were classified as Down-regulated (DOWN). All other genes, which did not meet these criteria for significant differential expression, were classified as Not differentially expressed (ND). Figure 3.2 shows the labeling's criteria.

The chosen thresholds of $p-adj < 0.1$ and $|log_2FC| \geq 1$ for classifying genes as UP or DOWN regulated are relatively common in differential expression analysis [69, 70]. A log2FC threshold of 1 or -1 represents a two-fold change in expression, often deemed biologically significant, while a p-adj threshold of 0.1 helps control false discovery rate in large-scale comparisons.

$$\text{Gene label} = \begin{cases} \text{UP} & \text{if } (\log_2 FC \geq 1) \text{ and } (p_{adj} < 0.1), \\ \text{DOWN} & \text{if } (\log_2 FC \leq -1) \text{ and } (p_{adj} < 0.1), \\ \text{ND} & \text{if } (-1 < \log_2 FC < 1) \text{ and } (p_{adj} > 0.1). \end{cases}$$

Figure 3.2: Criteria for labeling genes based on their differential expression.

In our study, we selected datasets 1, 4, 5, 6, 8 and 9 in Table 3.1 for training and 2, 3, 7 for validating. This means we have four studies with humans and two studies

with mice in host train set. Given the inherent biological variability and potential differences in experimental conditions, it is plausible that the same gene might exhibit different expression levels across these studies. To only label as UP or DOWN genes with a consistent pattern across different bacterial infection, we further established the following criteria:

For the human host studies:

1. If a gene obtained either UP or DOWN labels in two experiments and ND in the other one experiment, we assigned it the UP or DOWN label, respectively. This was based on the assumption that a consistent change in expression in two out of three experiments likely indicates a genuine biological effect.

2. If a gene maintained the same expression level (UP, DOWN, or ND) across all three experiments, we retained that expression level as the label for the gene. This consistency across experiments suggests a robust response.

3. Genes that did not conform to either of the above criteria were excluded from our analysis.

For the mice host experiments:

1. If a gene displayed the same expression level (UP, DOWN, or ND) in both experiments, we preserved that expression level as the label for the gene. This consistency across both experiments indicates a stable response.

2. Genes that did not meet the above criterion were discarded from our analysis.

This was implemented to ensure our analysis focused on genes with consistent expression patterns during a bacterial infection, thereby reducing potential noise and enhancing the reliability of our results.

3.2.7 Sequence Encoding

Sequence encoding is the process of transforming sequences into a format that can be used as input for machine learning models. While the RNA sequences contain rich information about the genes, they are not in a format that can be directly used by most machine learning models, which require numerical input. Therefore, we need the sequence of each gene from the genome file (for bacteria) and RNA transcripts (for hosts), and then transform these sequences into numerical features that capture the important characteristics of the sequences.

The reason we use the raw sequence from the genome file for bacteria is that bacteria do not undergo splicing, a process in which non-coding regions (introns) are removed from the pre-mRNA transcript and the remaining coding regions (exons) are joined together. Therefore, the sequence of a bacterial gene in the genome file is the same as the sequence of the corresponding RNA transcript.

On the other hand, for hosts, we use the sequence of the RNA transcripts rather than the raw sequence from the genome file. This is because hosts, being eukaryotes, do undergo splicing. The sequence of a gene in the genome file includes both exons and introns, but only the exons are included in the RNA transcript. Therefore, the

sequence of the RNA transcript provides a more accurate representation of the gene as it is expressed in the cell.

Since eukaryotic genes can have multiple transcripts, we selected the longest transcript for our analysis. This decision was based on the rationale that the longest transcript would likely contain the most comprehensive representation of the gene's potential information.

In this study, we employed MathFeature (version 1.0) [71], a software tool specifically designed for feature extraction from biological sequences. MathFeature was successfully employed in previous studies [72, 73], thereby demonstrating its effectiveness for this type of analysis.

A key consideration in our choice of sequence encoding methods was the ability to generate feature vectors of consistent length for all sequences, irrespective of the input sequence length. This ensures that the resulting feature vectors are compatible with the requirements of our machine learning models. The methods utilized in our study for feature extraction are detailed in Table 3.3. For k-mer and RC k-mer methods, we used $N = 4$. In the case of kGap, three distinct settings were tested: $(k = 2, x = 1, y = 2)$, $(k = 3, x = 1, y = 3)$, and $(k = 3, x = 2, y = 2)$.

After this step in the pipeline, each sequence is encoded into 1513 numerical features, and associated with a label as described in Section 3.2.6.

Table 3.3: Mathematical descriptors in our study. Explanations derived from [71, 74].

Descriptor	Explanation
Binary + Fourier	Represents nucleotides in binary, then applies Fourier transform for frequency analysis.
Z-curve + Fourier	Uses Z-curve representation of DNA sequences followed by Fourier transform.
Real + Fourier	Represents sequences in real numbers, then applies Fourier transform. G, A, C, and T are -0.5, -1.5, 0.5, and 1.5, respectively.
Integer + Fourier	Uses integer values for nucleotides, then applies Fourier transform. G, A, C, and T are 3, 2, 1, and 0, respectively.
EIIP + Fourier	Uses Electron-Ion Interaction Pseudopotential values for nucleotides, then applies Fourier transform. G, A, C, and T are 0.0806, 0.1260, 0.1340, and 0.1335, respectively.
Complex Number + Fourier	Represents sequences as complex numbers, then applies Fourier transform. G, A, C, and T are -1-j, 1+j, -1+j, and 1-j, respectively.
Atomic Number + Fourier	Uses atomic numbers, the total number of protons in each nucleotide then applies Fourier transform. G, A, C, and T are 78, 70, 58, and 66, respectively.
Shanon	Measures uncertainty or information content in DNA sequences.
Tsallis	A generalization of Shannon entropy, measures degree of disorder in sequences.
ORF Features or Coding Features	Features derived from Open Reading Frames or coding regions of sequences.
Fickett score	A measure used for prediction of protein-coding regions in DNA sequences.
k-mer	Represents sequences as overlapping substrings of length k.
Reverse Complement k-mer (RC k-mer)	Represents sequences and their reverse complements as overlapping substrings.
Xmer k-Spaced Ymer Composition Frequency (kGap)	Represents sequences based on the frequency of the pattern with X-mer follows k-gaps follows Y-mer.

3.3 Machine Learning Model Training

Upon preparing our dataset, the subsequent step in our analysis pipeline is to train machine learning models. Given the high dimensionality of our dataset, with each gene represented by 1513 features, we also incorporated dimensionality reduction methods into our pipeline.

In our study, we explored three dimensionality reduction methods: Principal Component Analysis (PCA) [75], Variational Autoencoders (VAE) [76], and Minimum Redundancy Maximum Relevance (mRMR) [77]. These methods were examined under

different conditions. Specifically, for PCA, we varied the number of components; for VAE, we changed the dimension of the latent space; and for mRMR, we adjusted the number of features selected. For each of these aspects across the three methods, we conducted tests at five different settings. This testing allowed us to explore the impact of these parameters on our models' performance.

PCA is a dimensionality reduction technique that transforms the original features into a new set of features, called principal components, which are linear combinations of the original features. VAE is a type of autoencoder, a neural network that is trained to reconstruct its input data, and it learns a lower-dimensional representation of the data in the process. mRMR is a dimensionality reduction method that aims to select features that are highly correlated with the target variable (maximum relevance) but have low correlation with each other (minimum redundancy).

Following the dimensionality reduction, we then chose three machine learning methods to determine the pairing of dimensionality reduction and machine learning model that generate the best performing model in terms of macro-average F1-score and macro-average AUROC. We evaluated Random Forests [78], XGBoost [49], and Light Gradient Boosted Machine (LightGBM) [79].

Our choice of models was influenced by the findings of Grinsztajn et al. [80], who demonstrated tree-based models such as Random Forests and XGBoost often outperform deep learning models on tabular data. This is due to their ability to effectively handle tabular data without relying on the invariances and spatial dependencies that

deep learning architectures are designed to capture. Additionally, tree-based models are more resistant to overfitting, especially when the data dimension is large relative to the number of examples—a common scenario in real-world tabular datasets.

3.4 Model hyper-parameters and implementation

3.4.1 Dimensionality Reduction

PCA

We implemented PCA using the PCA function in scikit-learn [81] (version 1.2.1), with the number of components (`n_components`) varying across five settings: 10, 20, 30, 40, and 50.

VAE

We implemented VAE using PyTorch [82] (version 2.0.1), spanning across 100 epochs with a learning rate of 0.002 using the Adam optimizer [83]. During the dimensionality reduction phase, the VAE's encoder transformed the original data into a latent space representation, which then served as input to our models. The VAE architecture, as shown in Figure 3.3 follows the design proposed by Wei and Ramsey [84]. It comprises an encoder with six hidden layers of sizes 1024, 512, 256, 128, 64, all using ReLU activation functions and Batch Normalization. Correspondingly, the decoder consists of six hidden layers of sizes 64, 128, 256, 512 and 1024 also employing ReLU

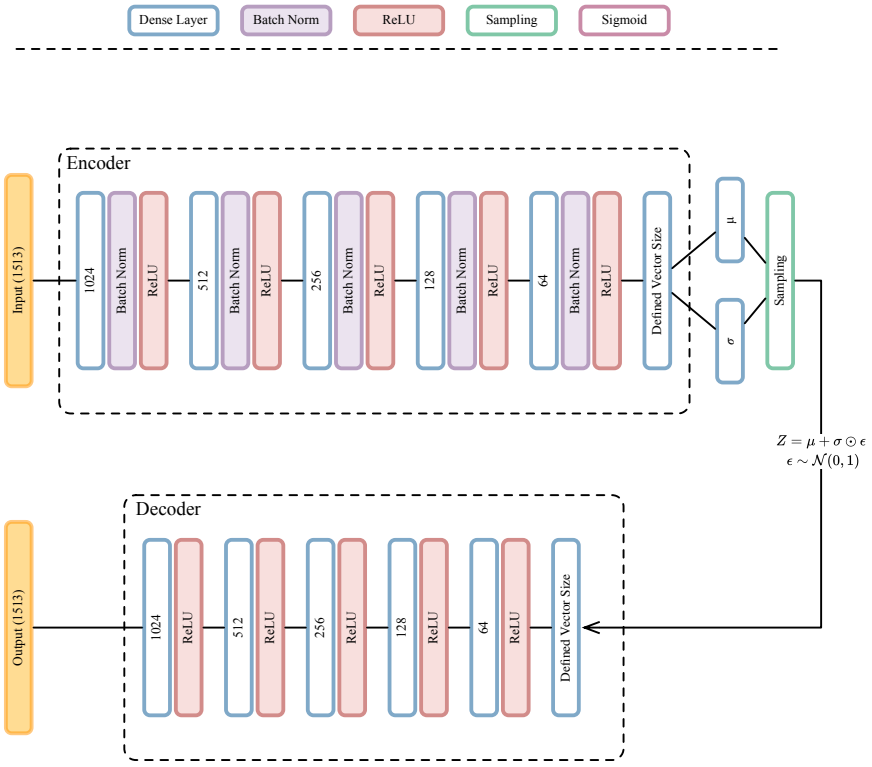

Figure 3.3: VAE architecture diagram. The vector sizes tested were 16, 32, 64, 128, and 256.

activation functions. The VAE was tested across five levels of features: 16, 32, 64, 128, and 256.

mRMR

We implemented mRMR feature selection method using the algorithm from a GitHub repository (https://github.com/smazzanti/mrmr). The mRMR was tested across five number of features: 300, 400, 500, 600, and 700.

3.4.2 Machine learning methods

Random Forest

The Random Forest model was generated using the scikit-learn Python package (version 1.2.1) [81]. This model was obtained with default parameters as per the package's settings.

XGBoost

The XGBoost model was generated via the XGBoost Python package (version 1.7.6) [49]. This model, too, was obtained with the package's preset default parameters.

LightGBM

The LightGBM model was generated using the LightGBM Python package (version 4.0.0) [79]. As with the other models, we used the default parameters provided by the package.

By keeping the parameters at their default settings across all three models, we maintained consistency during evaluation. The performance of each model was then assessed using the transformed datasets obtained from the PCA, VAE, and mRMR dimensionality reduction methods.

3.4.3 Performance Comparison

We integrated each machine learning method into our pipeline for the evaluation of all dimensionality reduction and machine learning method pairs. This allowed us to streamline the process from dimensionality reduction to model fitting and evaluation.

To assess model performance, we employed a stratified 10-fold cross-validation technique. This method ensured that each fold maintained the same proportions of sample labels as the complete set, thus providing a robust estimate of model performance.

Our evaluation metrics were macro-average F1-score and macro-average AUROC.

The macro-average F1-score is a metric that conveys the balance between precision and recall, averaged across all classes in a multi-class classification setting. Precision is defined as the ratio of true positive predictions to the sum of true positive and false positive predictions, as given by the formula:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall, also referred to as sensitivity, is the ratio of true positive predictions to the sum of true positive and false negative predictions, as given by the formula:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

The F1-score is the harmonic mean of precision and recall, given by the formula:

$$F1 - score = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1-score tries to find balance between precision and recall. The macro-average F1-score computes the F1-score for each class independently and then takes the average, which gives equal weight to all classes regardless of their size.

AUROC is a measure that evaluates the model's capability to distinguish between classes across various threshold settings. The Receiver Operating Characteristic (ROC) curve is a graphical plot that illustrates the true positive rate (recall) against the false positive rate (1-specificity) for different threshold values. The AUROC quantifies the overall ability of the model to discriminate between positive and negative instances across all threshold values, providing an overview of model performance that is independent of the decision threshold. The false positive rate is given by the formula:

$$\text{False Positive Rate} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$$

Macro-average AUROC is employed in multi-class classification settings to average the AUROC values obtained for each class against all others. In a multi-class classification, each class is considered as the positive class, and the rest are grouped as the negative class, and an AUROC is computed for each class in this manner. The macro-average AUROC is then the average of these AUROC values, calculated as:

$$\text{macro-average AUROC} = \frac{1}{N}\sum_{i=1}^{N}\text{AUROC}_i$$

where N is the number of classes, and AUROC_i is the AUROC for class i.

This approach gives equal weight to each class, irrespective of its size, which can be especially useful for our data since there are imbalances in class distribution. It

provides an equalized measure of the model's discriminative ability across all classes [85].

3.4.4 Hyper-Parameter Tuning

Upon comparing all combinations of dimensionality reduction and machine learning methods, we identified the most promising pair (dimensionality reduction and machine learning method combination) with the highest macro-average F1-score and macro-average AUROC. For these selected pairs, we proceeded to fine-tune their hyper-parameters to optimize their performance. We carried out this optimization using a Bayesian optimization approach via HyperOpt (version 0.2.7) [86]. This method of optimization allows for a more efficient search over the hyper-parameter space.

3.4.5 Model Training

Following the hyper-parameter tuning, we utilized the best parameters identified in that process to train the models on the complete training datasets. With these optimally-tuned models in hand, we then proceeded to evaluate their performance on the validation datasets.

3.5 Summary

We started our study by systematically collecting dual RNA-seq datasets from a variety of scientific literature sources. Our initial steps involved setting up a robust bioinformatics pipeline to process these raw data. This pipeline streamlined the transformation of raw sequencing reads into count tables. It included stages like quality control, facilitated by the fastp tool, followed by genome alignment, where we employed the STAR tool for host genomes and segemehl for pathogen genomes. The read counting was done by using the featureCounts tool.

Subsequently, we carried out a DGE analysis. This step was crucial for labeling the genes, using a predetermined set of criteria to ensure accuracy and consistency. As we transitioned to the machine learning phase, we have to convert the sequences into a format amenable for machine learning methods. This conversion was achieved using MathFeature, which transformed these sequences into numerical features.

For the dimensionality reduction phase, we tested three methods: PCA, VAE, and mRMR. Each method underwent evaluation under varied conditions to ascertain its efficacy. We selected Random Forests, XGBoost, and LightGBM as the machine learning methods to use, primarily because of their proven track record with tabular data and their resistance to overfitting.

To accurately assess model performance, we implemented a stratified 10-fold cross-validation technique using macro-average F1-score and macro-average AUROC. Additionally, we undertook a hyper-parameter tuning process to optimize the perfor-

mance of our models using a Bayesian optimization approach.

Chapter 4

Results and Discussion

In this chapter, we first present and discuss the outcomes derived from our bioinformatics pipeline. Subsequently, we provide the results of machine learning model selection and performed feature analysis to identify the variance captured by each principal component in PCA and the most important features in mRMR. Then, we assessed the performance of trained models on validation data. Finally, we perform Gene Ontology (GO) enrichment analysis to delve deeper into the biological significance of the predicted DEG.

4.1 Data

We used our bioinformatics pipeline to process the collected dual RNA-seq data to obtain log2 ratios and p-values per gene, and then used the criteria described in Section 3.2.6 to label each gene. The percentage of reads aligned, and the number of

Table 4.1: Percentage of reads aligned to host and pathogen genome for control (uninfected) samples and infected samples (genome alignment is described in Section 3.2.3) and number of genes assigned to each label per study (labeling is described in Section 3.2.6). Study No. corresponds to study numbers in Table 3.1. Studies 1, 4, 5, 6, 8, and 9 are selected for training and 2, 3, and 7 for validating. ND = not differentially expressed.

Study No.	Uniquely Mapped Reads (%)				Number of genes per label					
	Uninfected Samples		Infected Samples		Pathogen			Host		
	Pathogen	Host	Host	Pathogen	DOWN	ND	UP	DOWN	ND	UP
1	81.88%	84.35%	80.86%	0.02%	446	553	711	426	13541	818
2	92.72%	92.56%	63.65%	0.01%	743	500	168	4379	8328	4162
3	99.50%	91.14%	89.67%	0.64%	303	522	429	2683	10732	3193
4	92.23%	82.61%	82.13%	2.94%	416	1534	660	1937	8034	3249
5	68.11%	79.89%	66.12%	1.32%	221	617	240	811	14721	538
6	90.23%	73.62%	60.51%	0.09%	756	1483	738	2271	1347	2229
7	90.88%	72.49%	86.30%	2.92%	73	3430	193	1841	11380	989
8	98.58%	88.17%	88.98%	94.43%	1464	1347	1087	3577	6525	3716
9	57.71%	52.85%	51.26%	0.17%	194	631	103	1005	12522	1301

genes assigned to each of the three labels per study are shown in Table 4.1. Once individual study labels were obtained, we merged the study labels of the same organism as described in Section 3.2.6. Noteworthy, in study 8, reads for host and pathogen in infected samples were separated. Consequently, the percentages of uniquely mapped reads for each are independently determined.

The number of genes per each class for each organism in the final host and

Table 4.2: Number of genes per label in each host organism. ND = not differentially expressed.

Dataset	Name	DOWN	ND	UP
Train	*Homo sapiens*	86	2140	99
	Mus musculus	471	1359	921
	Total	557	3499	1020
Test	*Ictalurus punctatus*	4379	8328	4162
	Macaca fascicularis	2683	10732	3193
	Homo sapiens	1841	11380	989
	Total	8903	30440	8344

pathogen datasets is shown in Tables 4.2 and 4.3, respectively. In the training set in Table 4.2, the *Homo sapiens* and *Mus musculus* consists of three and two studies, respectively, and after applying the filtering criteria (labeling is described in Section 3.2.6), the number of genes per label got reduced.

4.2 Model Assessment

In our study, we assessed the performance of 45 models in total generated by combining dimensionality reduction algorithms and machine learning methods. To generate the classifiers, we used three dimensionality reduction algorithms, three machine learning methods, and we assessed the models performance using 10-fold cross-validation. A detailed breakdown of these comparisons can be found in Table 4.4 and

Table 4.3: Number of genes per label in each pathogen organism. ND = not differentially expressed.

Dataset	Name	DOWN	ND	UP
Train	*Porphyromonas gingivalis*	446	553	711
	Mycobacterium tuberculosis H37Rv	416	1534	660
	Haemophilus influenzae Fi176, Hi176	221	617	240
	Bordetella pertussis Tohama I	756	1483	738
	Pseudomonas aeruginosa PAO1	1464	1347	1087
	Staphylococcus aureus subsp. aureus NCTC 8325	194	631	103
	Total	3497	6165	3539
Test	*Yersinia ruckeri* strain YZ	743	500	168
	Streptococcus pyogenes strain MGAS2221	303	522	429
	Mycobacterium tuberculosis variant bovis BCG str. ATCC 35733	73	3430	193
	Total	1119	4452	790

Table 4.5 for host and pathogen, respectively.

In the evaluation of our classifiers, models were primarily assessed based on their accuracy, macro-average F1-score and macro-average AUROC. For the host dataset, the leading configurations were XGBoost combined with mRMR (500 features), Random Forest coupled with mRMR (400 features), and LightGBM with mRMR (600 features). In the context of the pathogen dataset, the standout configurations were XGBoost paired with PCA (with the first 30 components), Random Forest with PCA (with the first 20 components), and LightGBM combined with mRMR (400 features). Subsequently, a hyper-parameter optimization was conducted on these configurations

Table 4.4: Cross-validation results of different classifiers for host gene expression prediction. Best performing model for each classifier is highlighted.

Classifier	Dimensionality Reduction Algorithm	Feature Vector Size	Accuracy	Macro-Average F1-Score	Macro-Average AUROC
XGBoost	VAE	16	0.6753 ± 0.0083	0.3813 ± 0.0138	0.6145 ± 0.0167
		32	0.6684 ± 0.0161	0.3614 ± 0.0253	0.6109 ± 0.0156
		64	0.6682 ± 0.0141	0.3716 ± 0.0165	0.6077 ± 0.0261
		128	0.6740 ± 0.0128	0.3811 ± 0.0198	0.6172 ± 0.0229
		256	0.6698 ± 0.0146	0.3801 ± 0.0164	0.6127 ± 0.0254
	PCA	10	0.6895 ± 0.0177	0.4075 ± 0.0269	0.6483 ± 0.0264
		20	0.6992 ± 0.0138	0.4236 ± 0.0203	0.6891 ± 0.0105
		30	0.7053 ± 0.0146	0.4268 ± 0.0171	0.6966 ± 0.0186
		40	0.7094 ± 0.0122	0.4304 ± 0.0134	0.7005 ± 0.0115
		50	0.7055 ± 0.0110	0.4180 ± 0.0144	0.6978 ± 0.0193
	MRMR	300	0.7126 ± 0.0088	0.4334 ± 0.0127	0.7078 ± 0.0221
		400	0.7159 ± 0.0109	0.4337 ± 0.0151	0.7144 ± 0.0225
		500	0.7177 ± 0.0114	0.4361 ± 0.0181	0.7202 ± 0.0171
		600	0.7155 ± 0.0121	0.4306 ± 0.0194	0.7133 ± 0.0262
		700	0.7161 ± 0.0141	0.4352 ± 0.0218	0.7118 ± 0.0190
Random Forest	VAE	16	0.6783 ± 0.0117	0.3653 ± 0.0152	0.6113 ± 0.0173
		32	0.6795 ± 0.0127	0.3663 ± 0.0196	0.6126 ± 0.0234
		64	0.6757 ± 0.0125	0.3584 ± 0.0171	0.6127 ± 0.0267
		128	0.6742 ± 0.0168	0.3681 ± 0.0209	0.6180 ± 0.0236
		256	0.6797 ± 0.0107	0.3698 ± 0.0183	0.6173 ± 0.0224
	PCA	10	0.7047 ± 0.0120	0.3904 ± 0.0136	0.6592 ± 0.0208
		20	0.7096 ± 0.0077	0.3853 ± 0.0134	0.6886 ± 0.0264
		30	0.7090 ± 0.0055	0.3732 ± 0.0114	0.6895 ± 0.0245
		40	0.7055 ± 0.0050	0.3586 ± 0.0155	0.6908 ± 0.0243
		50	0.7029 ± 0.0030	0.3464 ± 0.0085	0.6917 ± 0.0267
	MRMR	300	0.7136 ± 0.0082	0.3928 ± 0.0154	0.6905 ± 0.0222
		400	0.7143 ± 0.0099	0.3943 ± 0.0164	0.6946 ± 0.0262
		500	0.7114 ± 0.0080	0.3891 ± 0.0158	0.6936 ± 0.0224
		600	0.7118 ± 0.0066	0.3839 ± 0.0157	0.6940 ± 0.0231
		700	0.7122 ± 0.0077	0.3874 ± 0.0107	0.6919 ± 0.0202
LightGBM	VAE	16	0.6726 ± 0.0123	0.3660 ± 0.0181	0.6215 ± 0.0183
		32	0.6734 ± 0.0111	0.3600 ± 0.0161	0.6100 ± 0.0142
		64	0.6730 ± 0.0133	0.3629 ± 0.0230	0.6167 ± 0.0225
		128	0.6726 ± 0.0092	0.3617 ± 0.0160	0.6214 ± 0.0229
		256	0.6751 ± 0.0148	0.3750 ± 0.0193	0.6206 ± 0.0179
	PCA	10	0.6962 ± 0.0185	0.4043 ± 0.0238	0.6685 ± 0.0236
		20	0.7078 ± 0.0148	0.4332 ± 0.0207	0.6962 ± 0.0136
		30	0.7108 ± 0.0102	0.4289 ± 0.0112	0.7027 ± 0.0160
		40	0.7090 ± 0.0136	0.4215 ± 0.0164	0.7106 ± 0.0170
		50	0.7122 ± 0.0100	0.4234 ± 0.0150	0.7060 ± 0.0123
	MRMR	300	0.7139 ± 0.0092	0.4298 ± 0.0141	0.7171 ± 0.0282
		400	0.7157 ± 0.0110	0.4318 ± 0.0191	0.7223 ± 0.0253
		500	0.7177 ± 0.0075	0.4309 ± 0.0165	0.7191 ± 0.0246
		600	0.7205 ± 0.0107	0.4382 ± 0.0126	0.7244 ± 0.0222
		700	0.7169 ± 0.0106	0.4310 ± 0.0167	0.7243 ± 0.0256

Table 4.5: Cross-validation results of different classifiers for pathogen gene expression prediction. Best performing model for each classifier is highlighted.

Classifier	Dimensionality Reduction Algorithm	Feature Vector Size	Accuracy	Macro-Average F1-Score	Macro-Average AUROC
XGBoost	VAE	16	0.4765 ± 0.0137	0.3919 ± 0.0249	0.5976 ± 0.0235
		32	0.4718 ± 0.0108	0.3999 ± 0.0166	0.6048 ± 0.0125
		64	0.4629 ± 0.0172	0.3896 ± 0.0281	0.5881 ± 0.0299
		128	0.4735 ± 0.0166	0.4045 ± 0.0260	0.6101 ± 0.0241
		256	0.4734 ± 0.0178	0.4104 ± 0.0276	0.6092 ± 0.0223
	PCA	10	0.4767 ± 0.0124	0.4039 ± 0.0142	0.6199 ± 0.0084
		20	0.4897 ± 0.0083	0.4208 ± 0.0087	0.6344 ± 0.0069
		30	0.4938 ± 0.0103	0.4282 ± 0.0129	0.6360 ± 0.0097
		40	0.4887 ± 0.0116	0.4247 ± 0.0142	0.6292 ± 0.0085
		50	0.4907 ± 0.0087	0.4247 ± 0.0110	0.6301 ± 0.0068
	MRMR	300	0.4910 ± 0.0055	0.4232 ± 0.0084	0.6275 ± 0.0065
		400	0.4908 ± 0.0102	0.4212 ± 0.0127	0.6288 ± 0.0112
		500	0.4909 ± 0.0058	0.4226 ± 0.0078	0.6323 ± 0.0071
		600	0.4938 ± 0.0113	0.4268 ± 0.0121	0.6309 ± 0.0063
		700	0.4926 ± 0.0129	0.4250 ± 0.0150	0.6322 ± 0.0070
Random Forest	VAE	16	0.4719 ± 0.0314	0.4060 ± 0.0335	0.6080 ± 0.0356
		32	0.4705 ± 0.0245	0.3944 ± 0.0367	0.6023 ± 0.0281
		64	0.4705 ± 0.0180	0.3961 ± 0.0337	0.6061 ± 0.0291
		128	0.4617 ± 0.0284	0.4015 ± 0.0254	0.6023 ± 0.0272
		256	0.4790 ± 0.0199	0.4160 ± 0.0188	0.6174 ± 0.0194
	PCA	10	0.4863 ± 0.0105	0.4143 ± 0.0128	0.6314 ± 0.0083
		20	0.5122 ± 0.0078	0.4312 ± 0.0118	0.6543 ± 0.0108
		30	0.5104 ± 0.0089	0.4228 ± 0.0118	0.6498 ± 0.0098
		40	0.5100 ± 0.0109	0.4158 ± 0.0155	0.6513 ± 0.0082
		50	0.5029 ± 0.0075	0.4051 ± 0.0087	0.6486 ± 0.0078
	MRMR	300	0.5062 ± 0.0075	0.4081 ± 0.0112	0.6411 ± 0.0089
		400	0.5073 ± 0.0082	0.4069 ± 0.0123	0.6416 ± 0.0090
		500	0.5037 ± 0.0102	0.4022 ± 0.0132	0.6434 ± 0.0088
		600	0.5018 ± 0.0079	0.4006 ± 0.0122	0.6433 ± 0.0073
		700	0.5025 ± 0.0080	0.3978 ± 0.0117	0.6389 ± 0.0097
LightGBM	VAE	16	0.4899 ± 0.0071	0.4040 ± 0.0133	0.6271 ± 0.0109
		32	0.4887 ± 0.0258	0.3941 ± 0.0478	0.6123 ± 0.0402
		64	0.4742 ± 0.0215	0.3741 ± 0.0430	0.5947 ± 0.0314
		128	0.4847 ± 0.0167	0.4042 ± 0.0247	0.6194 ± 0.0214
		256	0.4763 ± 0.0182	0.3870 ± 0.0286	0.6071 ± 0.0249
	PCA	10	0.4942 ± 0.0097	0.4103 ± 0.0134	0.6330 ± 0.0074
		20	0.4995 ± 0.0070	0.4181 ± 0.0078	0.6461 ± 0.0058
		30	0.5032 ± 0.0080	0.4213 ± 0.0093	0.6460 ± 0.0102
		40	0.5016 ± 0.0108	0.4193 ± 0.0145	0.6443 ± 0.0091
		50	0.5042 ± 0.0112	0.4227 ± 0.0139	0.6438 ± 0.0086
	MRMR	300	0.5044 ± 0.0119	0.4215 ± 0.0162	0.6439 ± 0.0088
		400	0.5038 ± 0.0094	0.4200 ± 0.0092	0.6406 ± 0.0083
		500	0.5045 ± 0.0092	0.4226 ± 0.0124	0.6446 ± 0.0065
		600	0.5061 ± 0.0122	0.4234 ± 0.0154	0.6461 ± 0.0097
		700	0.5050 ± 0.0090	0.4219 ± 0.0113	0.6432 ± 0.0043

Table 4.6: Cross-validation results of candidate classifiers (highlighted on Table 4.4) with optimal hyper-parameters on host dataset.

Classifier	**Accuracy**	**Macro-Average F1-Score**	**Macro-Average AUROC**
XGBoost with mRMR (500 features)	0.7021 ± 0.0164	0.4462 ± 0.0228	0.6916 ± 0.0206
Random Forest with mRMR (400 features)	**0.6322 ± 0.0238**	**0.4985 ± 0.0215**	**0.7106 ± 0.0182**
LightGBM with mRMR (600 features)	0.6976 ± 0.0126	0.4537 ± 0.0244	0.6899 ± 0.0220

Table 4.7: Cross-validation results of candidate classifiers (highlighted on Table 4.5) with optimal hyper-parameters on pathogen dataset.

Classifier	**Accuracy**	**Macro-Average F1-Score**	**Macro-Average AUROC**
XGBoost with PCA (30 components)	0.5086 ± 0.0088	0.4501 ± 0.0107	0.6516 ± 0.0104
Random Forest with PCA (20 components)	**0.4939 ± 0.0097**	**0.4680 ± 0.0108**	**0.6614 ± 0.0073**
LightGBM with mRMR (600 features)	0.5066 ± 0.0101	0.4446 ± 0.0125	0.6447 ± 0.0062

using HyperOpt. Tables 4.6 and 4.7 present the results of hyper-parameter optimization on classifiers on host and pathogen datasets, respectively. Our final selections were Random Forest with mRMR (600 features) for the host and Random Forest with PCA (with the first 20 components) for the pathogen. The optimal parameter's spaces for host and pathogen classifiers are provided in Table 4.8.

Table 4.8: Optimal hyper-parameters for Random Forest classifiers on host and pathogen datasets.

Parameter	Values	
	Host Classifier	**Pathogen Classifier**
bootstrap	FALSE	TRUE
class_weight	balanced	balanced
max_depth	11	14
max_features	sqrt	sqrt
min_samples_leaf	13	3
min_samples_split	2	17
n_estimators	100	400

4.3 Feature Analysis

Before proceeding with the validation tests, we performed the feature analysis for both dimensionality reduction algorithms to identify the variance captured by each principal component in PCA and the most important features in mRMR.

4.3.1 Feature Importance Analysis

Pathogen model - Random Forest and PCA

To effectively illustrate how the variance in the pathogen data is distributed among the different principal components, we utilize a scree plot as shown in Figure 4.1. The scree plot graphically displays the fraction of total variance explained by each

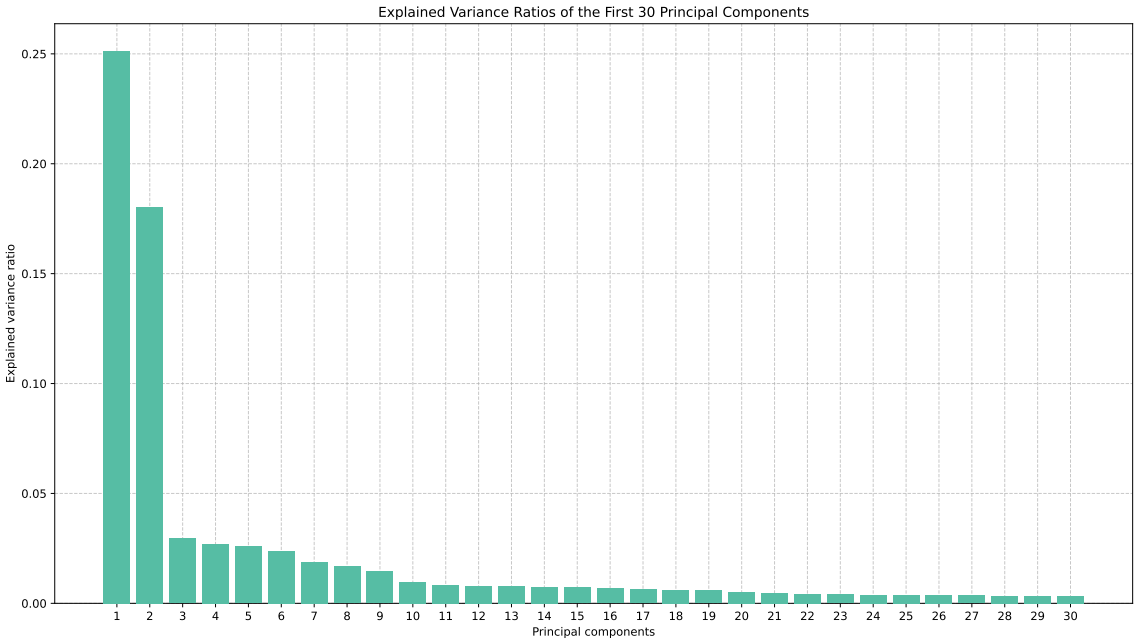

Figure 4.1: Scree plot for the first 30 principal components from the pathogen dataset using PCA.

principal component. It aids in determining the optimal number of components to retain for further analysis.

Figure 4.1 illustrates that the first two principal components collectively account for approximately 35% of the total variance in the pathogen data, suggesting substantial redundancy or correlation among the original features. Beyond the second component, there is a noticeable diminishing return in variance explained. Specifically, from the fifth to the thirtieth component, only an additional 20% of the variance is captured.

Upon training our model using the parameters identified during hyper-parameter tuning (detailed in Section 3.3), we evaluated the importance of the chosen 20 compo-

nents. We used the `feature_importances_` attribute of the Random Forest Classifier for this purpose. The components exhibited an average importance score of 0.05, with a standard deviation of 0.01063. This narrow spread in importance scores indicates a consistent value being assigned to each component, implying that they all play nearly equivalent roles in determining the model's predictions. This observation further supports the idea that our selected components effectively capture a comprehensive range of relevant information from the dataset.

Following the feature importance analysis, we further investigated the relevance of each of the 20 selected components through permutation importance analysis through applying the `permutation_importance` function from scikit-learn, utilizing the macro-average F1-score for scoring.

Figure 4.2 reveals the importances scores, with 10 components exhibiting positive scores and the remaining 10 showcasing negative scores. Positive scores in this context imply that the corresponding components possess a substantial influence on the predictive accuracy of the model; a permutation in the values of these components leads to a noticeable decrease in the macro-average F1-score, underlining their importance in the model. The components with positive importance scores can be viewed as integral factors underpinning the predictive capacity of the model. Their values hold meaningful information that allows the model to steer towards accurate predictions.

Conversely, the components that received negative scores in the permutation im-

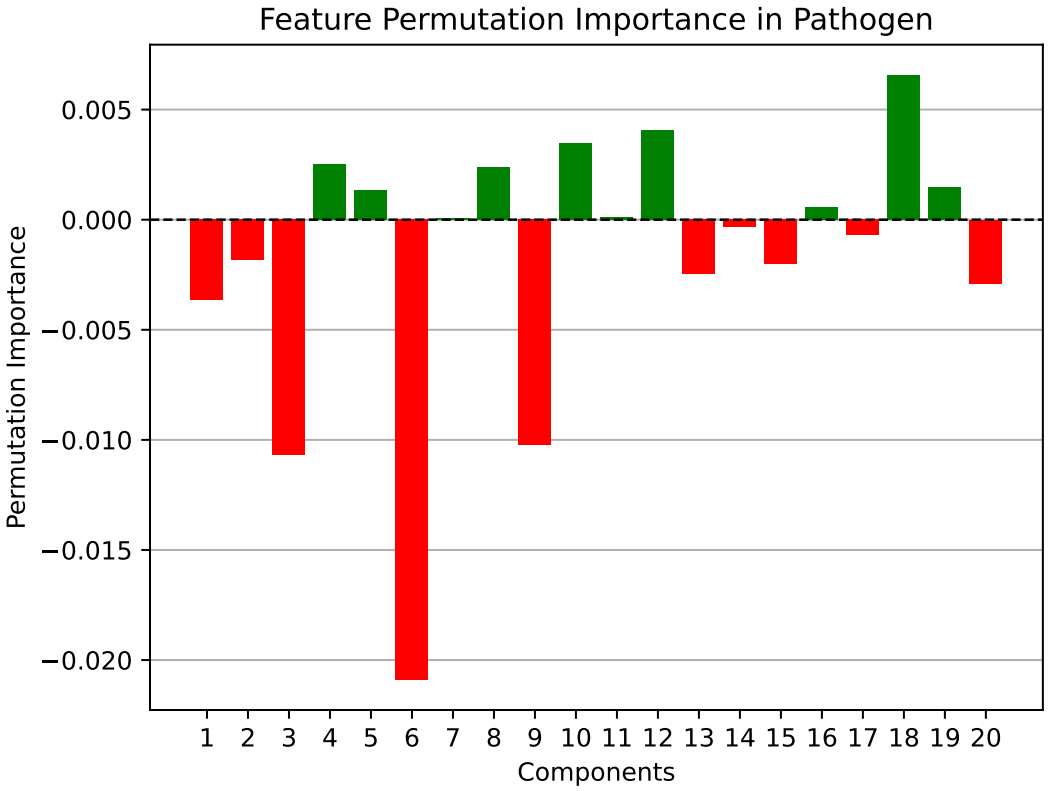

Figure 4.2: Permutation importance scores of the first 20 principal components in pathogen dataset. Each bar represents a component's influence on the model's macro-average F1-score.

portance analysis signal a contrary phenomenon. A permutation in the values of these components tends to increase the macro-average F1-score, suggesting that these components, in their original state, might have been introducing noise to the model, potentially misguiding the predictions. These components appear to have a counter-productive effect on the model's predictive accuracy, necessitating a critical examination to discover whether keeping them in the model is beneficial.

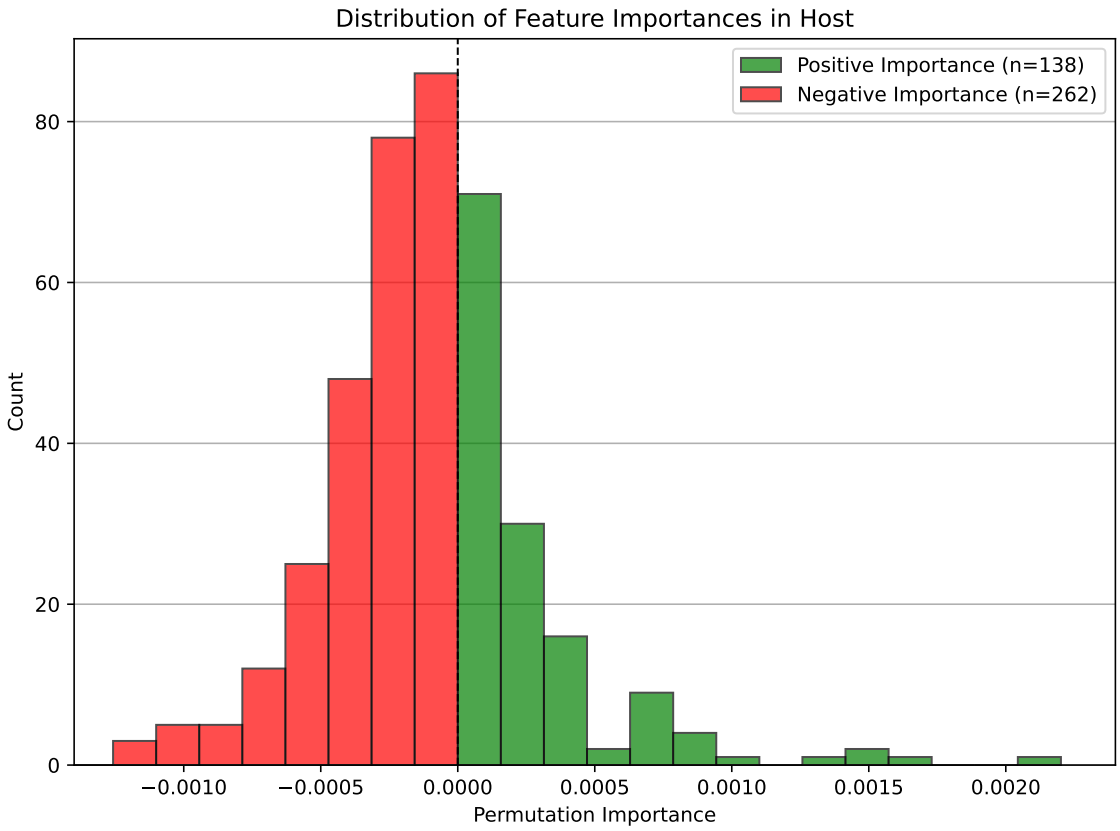

Figure 4.3: Distribution of feature importance scores of 400 features obtained from mRMR in host dataset.

Host model - Random Forest and mRMR

To delve deeper into the impact of the 400 features curated through the mRMR dimensionality reduction process, we applied the `permutation_importance` function from scikit-learn, utilizing the macro-average F1-score for scoring.

The distribution of the permutation importance scores for all the features is depicted in Figure 4.3. The analysis returned both positive and negative scores, with 138 features having a positive score and 262 yielding a negative score.

Table 4.9: Top 20 mRMR features with the highest importance scores for the host classifier.

Feature	Importance Score	Descriptor
ACGT	0.002201	k-mer
AG---CG	0.001648	kGap
A---CTC	0.001502	kGap
AG---AA	0.001433	kGap
A--AT	0.001364	kGap
CA---CT	0.000975	kGap
A---AAT	0.000933	kGap
CA---CG	0.000861	kGap
A---CAG	0.000793	kGap
CA---TG	0.000788	kGap
ATGC	0.000756	RC k-mer
A---CCT	0.000746	kGap
ATAA	0.000704	RC k-mer
ACT	0.000698	k-mer
AGAG	0.000696	k-mer
A---CGG	0.000681	kGap
AC---CT	0.000676	kGap
ACAC	0.000667	k-mer
A--GG	0.000647	kGap
ATGT	0.000592	k-mer

Positive scores in this context suggest features that actively enhance the model's performance, while the negative scores tend to indicate features that could potentially be impairing the predictive power.

Table 4.9 shows the top 20 mRMR features, ranked by their importance scores. The table also lists the descriptor groups to which each feature belongs.

Table 4.10: Cross-validation results for the host classifier, using features with positive importance scores (138 features) and those in the original set of 400 features.

Classifier	Accuracy	Macro-Average F1-Score	Macro-Average AUROC
Random Forest with mRMR (138 features)	0.6146 ± 0.0195	0.4624 ± 0.0230	0.6908 ± 0.0203
Random Forest with mRMR (400 features)	0.6322 ± 0.0238	0.4985 ± 0.0215	0.7106 ± 0.0182

Addressing Negative Permutation Importance Scores

In response to the results derived from the permutation importance analysis, an explorative step was undertaken to understand the implications of the negatively scored components on the predictive models. This involved removing the components and features with negative scores in PCA and mRMR to observe any potential enhancements in the model's performance. Contrary to expectations, removing these components did not improve performance, as shown in Table 4.10 and 4.11. This incident emphasizes the intricate and potentially non-linear relationships these features/components may share with others in influencing the predictive outcomes, where information that seems to have a negative influence in isolation can play a supportive role in the context of other features/components.

4.4 Model Assessment

In each ROC plot in Figures 4.4 and 4.5, a shaded area depicts the range of ROC curves observed across different folds, showcasing the area between the maximum and

Table 4.11: Cross-validation results for the pathogen classifier, using components with positive importance scores (10 components) and those in the original set of 20 components.

Classifier	Accuracy	Macro-Average F1-Score	Macro-Average AUROC
Random Forest with PCA (10 components)	0.4634 ± 0.0076	0.4372 ± 0.0089	0.6297 ± 0.0080
Random Forest with PCA (20 components)	0.4939 ± 0.0097	0.4680 ± 0.0108	0.6614 ± 0.0073

minimum ROC curves for each class. This visual depiction helps in understanding the extent of performance variance across different data subsets and provides a measure of the classifier's stability.

4.4.1 Training Performance Analysis for Host

Figure 4.4 illustrates the ROC curves during cross-validation for all three classes along with the macro-average, with a relatively narrow shaded area, indicating a consistent performance across different folds.

- The macro-average ROC curve shows an Area Under Curve (AUC) of 0.71, reflecting a good general performance of the classifier.
- Classification of ND genes achieved an AUC of 0.71, mirroring the macro-average, and denoting competent classifier performance in this class.
- Classification of UP genes achieved an AUC of 0.74, indicating the classifier can distinguish genes in this class.

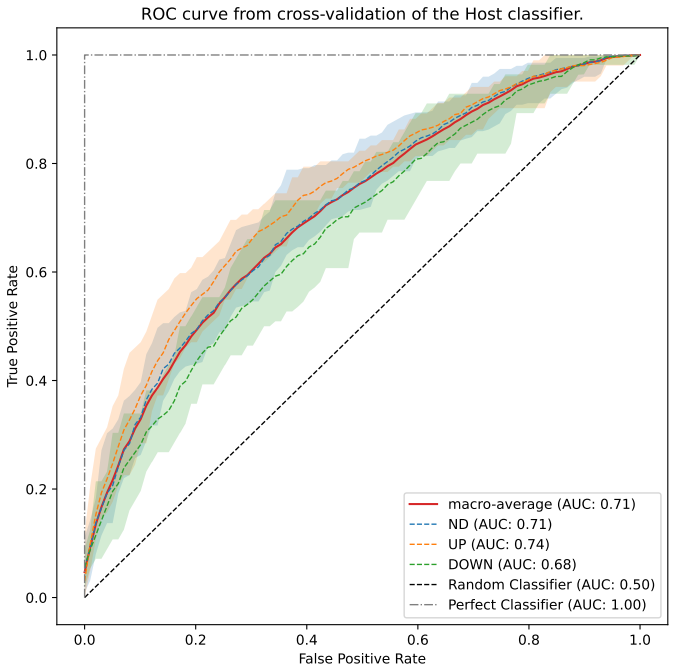

Figure 4.4: ROC curve from cross-validation results of the Host classifier.

- Classification of DOWN genes achieved an AUC of 0.68, which, although slightly lower than the AUC obtained for the other two classes, still denotes a reasonable classifier performance.

The closeness of the ROC curves and the narrow shaded area signify a stable performance of the host classifier across the three gene expression classes.

4.4.2 Training Performance Analysis for Pathogen

Figure 4.5 depicts the ROC curves during cross-validation for all three classes and the macro-average exhibit close proximity to each other, with a relatively narrow shaded area indicating a consistent performance across different folds.

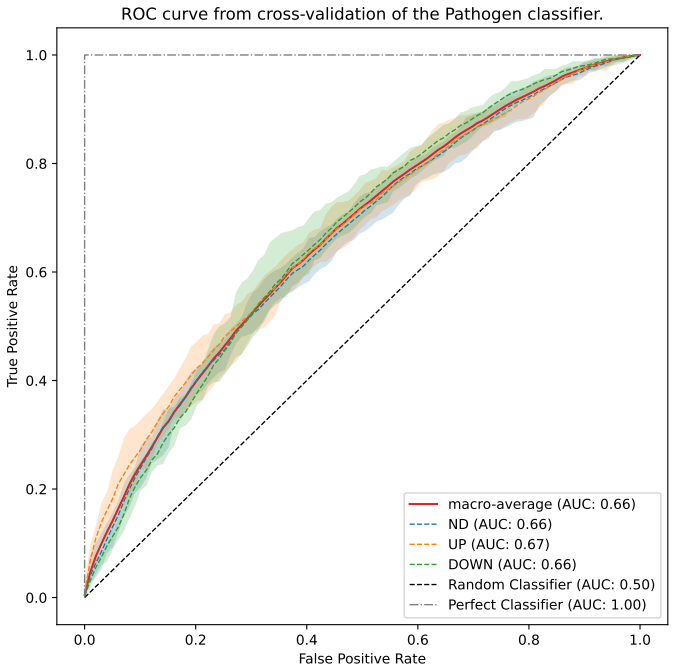

Figure 4.5: ROC curve from cross-validation results of the Pathogen classifier.

- The macro-average ROC curve presents an AUC of 0.66, reflecting a fair general performance of the classifier.
- Classification of ND genes achieved an AUC of 0.66, mirroring the macro-average, and signifying a decent classifier performance in this class.
- Classification of ND genes achieved an AUC of 0.67, indicating an enhanced ability of the classifier to distinguish genes in this class.
- Classification of ND genes achieved an AUC of 0.66, consistent with the macro-average and ND class, demonstrating a reasonable classifier performance.

The closeness of the ROC curves and the narrow shaded area signify a stable

performance of the pathogen classifier across the three gene expression classes, albeit at a modest level.

The performance of the classifier on the host data is better compared to the performance on the pathogen data. There potential reason for this observation is that we used data from four human (*Homo sapiens*) and two mouse (*Mus musculus*) studies and chose genes that showed clear patterns across these studies (as discussed in Section 3.2.6). This means the host classifier had clear and consistent information to learn from. On the other hand, the pathogen classifier used mixed data from six different bacteria. Mixing all this data might have made it harder for the pathogen classifier to find clear patterns, which could explain why it did not do as well as the host classifier.

4.4.3 Test Performance Analysis for Host

For the independent test data, genes were labeled based on a single study, and thus, there are more genes per each label for the hosts in the test data (as shown in Table 4.2). As the host genes were not filtered for consistency across studies, it is likely that the host gene labels in the test data are noisier than those in the training data.

Homo sapiens

Figure 4.6 shows that the classifier performs better than random guessing, as the curve lies above the diagonal line representing random classification. The macro-average

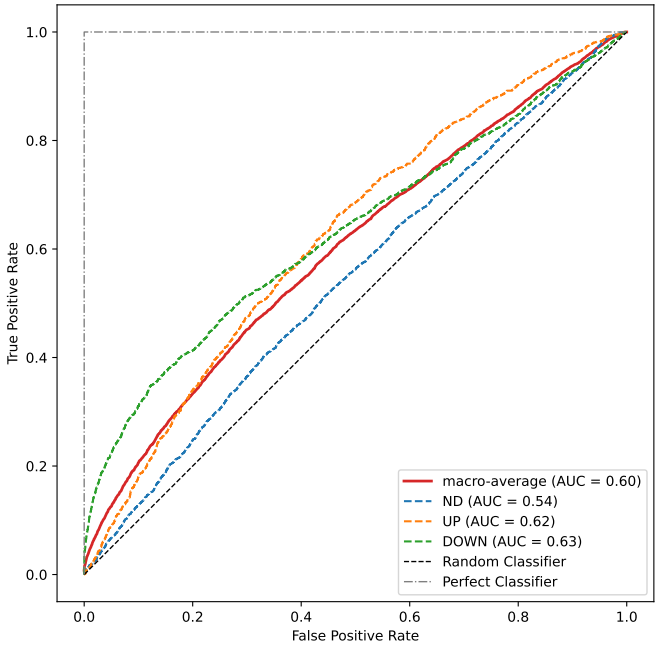

Figure 4.6: ROC Curve of *Homo sapiens*.

ROC curve, which provides an average performance measure across all classes, has an AUC of 0.60. This suggests that the classifier's overall performance for predicting gene expression levels during a bacterial infection in *Homo sapiens* is better than random guessing. When considering the individual classes:

- Classification of ND genes achieved an AUC of 0.54, indicating a performance slightly better than random.
- Classification of UP genes achieved an AUC of 0.62, aligning closely with the macro-average.
- Classification of DOWN genes achieved an AUC of 0.63, which is the highest among the three classes.

In summary, the Host classifier is able to classify genes according to their expression level during infection substantially better than a random classifier. As the classifier was trained with human data from other studies, this result suggests that the model is able to generalize to an organism seen during training responding to a different bacterial infection.

Macaca fascicularis

Figure 4.7 depicts the ROC curve for *Macaca fascicularis* (crab-eating macaque) and suggests a performance closer to the diagonal line representing random classification than the model performance for the other two hosts (human and channel catfish) used for validation. This visual observation aligns with the provided AUC values, indicating that the classifier's performance for this species is slightly better than random guessing.

The macro-average ROC curve, representing an average performance across all classes, has an AUC of 0.51. This demonstrates that the classifier's overall ability to predict gene expression levels in *Macaca fascicularis* is slightly above random classification. When examining the individual classes:

- Classification of ND genes achieved an AUC of 0.52, reflecting a performance just a notch above random.
- Classification of UP genes achieved an AUC of 0.54. This suggests a slight edge in the classifier's capability to distinguish up-regulated genes relative to

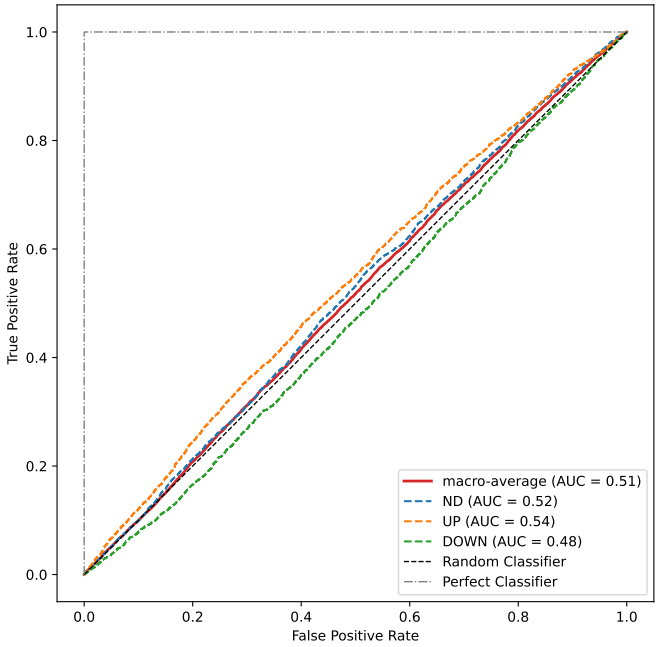

Figure 4.7: ROC Curve of *Macaca fascicularis*.

the other categories.

- Classification of DOWN genes achieved an AUC of 0.48, which is worse than random guessing. This indicates that the classifier struggles to differentiate down-regulated genes effectively.

The lower macro-average AUC achieved by the host classifier for *Macaca fascicularis* is surprising as we have hypothesized that prediction of expression levels in the *Macaca fascicularis* would be similar to that in human.

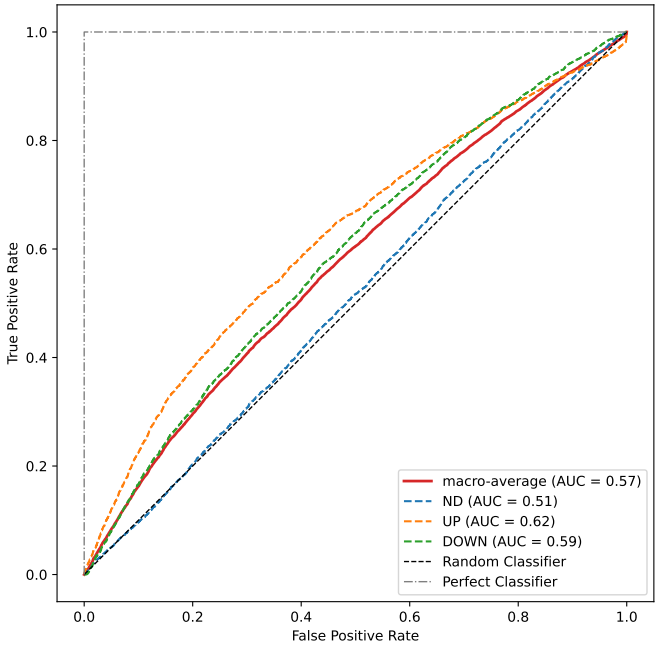

Figure 4.8: ROC Curve of *Ictalurus punctatus*.

Ictalurus punctatus

Figure 4.8 suggests that the classifier's performance is above the diagonal line indicative of random classification for *Ictalurus punctatus* (channel catfish). The macro-average ROC curve, reflecting an average performance metric across all classes, yields an AUC of 0.57. This suggests that the classifier's overall proficiency in predicting gene expression levels in *Ictalurus punctatus* is above random guessing. Delving into the performance for individual classes:

- Classification of ND genes achieved an AUC of 0.51, which is narrowly above the threshold for random classification. This indicates modest performance in this category.

- Classification of UP genes achieved an AUC of 0.62, suggesting the classifier has an enhanced ability to discern up-regulated genes compared to the other classes.

- Classification of DOWN genes achieved an AUC of 0.59, situating it between the ND and UP classes in terms of performance.

To summarize, the ROC curve for *Ictalurus punctatus* implies a performance that is superior to random guessing. Among the gene expression categories, the classifier exhibits its best performance for the up-regulated and down-regulated genes. This achieved classification performance is quite remarkable if one considers that the training data only contained mammalian hosts. This result suggests that responses to some bacterial infections are similar among diverse hosts.

4.4.4 Test Performance Analysis for Pathogen

***Mycobacterium tuberculosis* (*M. tuberculosis*)**

Figure 4.9 visualizes the ROC curve for *M. tuberculosis* and suggests a performance better than random classification. This visual observation is further nuanced by the AUC values. The macro-average ROC curve reveals an AUC of 0.55, suggesting a performance above random classification. Among the individual classes:

- Classification of ND genes achieved an AUC of 0.63, indicating a relatively good ability to predict this category.

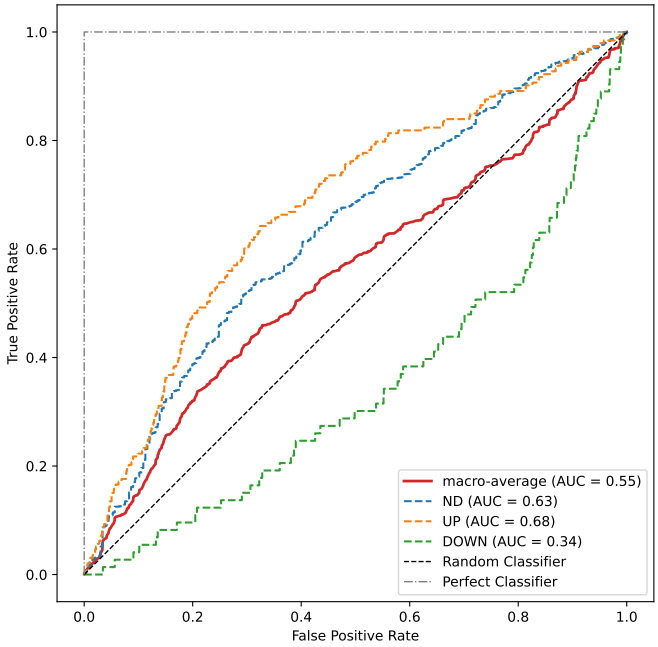

Figure 4.9: ROC Curve of *M. tuberculosis*.

- Classification of UP genes achieved an AUC of 0.68, suggesting a decent proficiency in distinguishing up-regulated genes.

- Classification of DOWN genes achieved an AUC of 0.34, implying significant challenges in effectively classifying these genes.

To summarize, the ROC curve for *M. tuberculosis* implies a performance that is superior to random guessing. Among the gene expression categories, the classifier exhibits its best performance for the up-regulated genes.

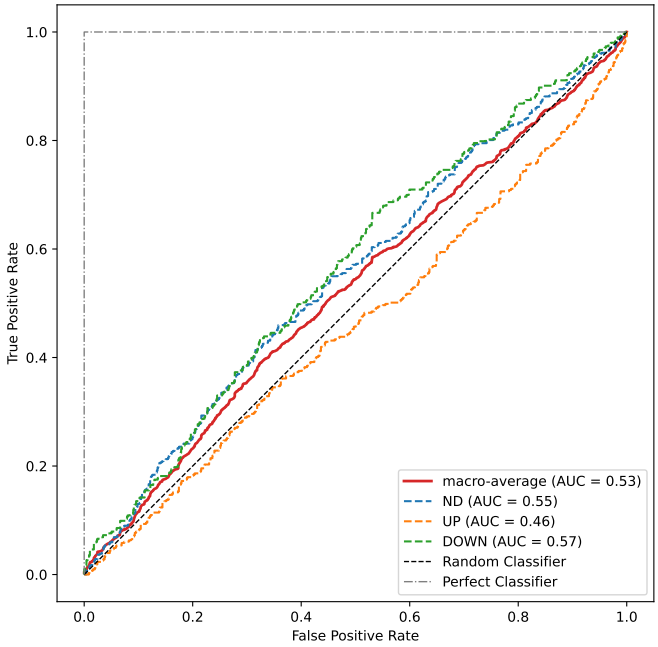

Figure 4.10: ROC Curve of *S. pyogenes*.

***Streptococcus pyogenes* (*S. pyogenes*)**

Figure 4.11 displays a performance above random guessing for *S. pyogenes*, albeit not significantly. The macro-average AUC stands at 0.53, further confirming this observation. Delving into individual classes:

- Classification of ND genes achieved an AUC of 0.55, slightly better than the macro-average.
- Classification of UP genes achieved an AUC of 0.46, underperform, indicating difficulties in differentiating up-regulated genes for this bacterium.
- Classification of DOWN genes achieved an AUC of 0.57, positioning it as the

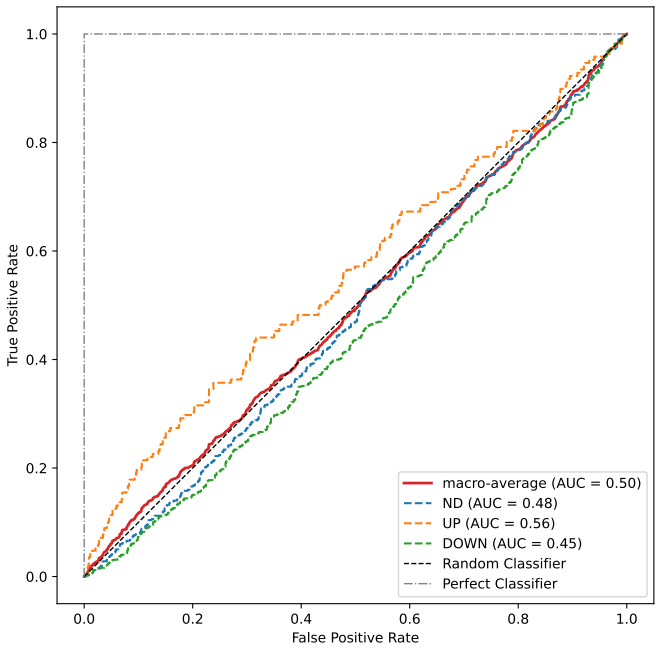

Figure 4.11: ROC Curve of *Y. ruckeri*.

best-performing class for this bacterium, albeit only marginally.

To summarize, the ROC curve for *S. pyogenes* suggests a performance that is slightly better than random guessing. Among the gene expression categories, the classifier exhibits its best performance for the down-regulated genes, with challenges evident in identifying up-regulated genes.

Y. ruckeri

Figure 4.11 suggests a performance close to random classification for *Y. ruckeri*. Examining the individual classes:

- Classification of ND genes achieved an AUC of 0.48, slightly below random

guessing.

- Classification of UP genes achieved an AUC of 0.56, suggesting a modest ability to identify up-regulated genes.
- Classification of DOWN genes achieved an AUC of 0.45, hint at challenges in distinguishing down-regulated genes.

To summarize, the ROC curve for *Y. ruckeri* essentially aligns with random guessing in terms of performance. Among the gene expression categories, the classifier exhibits its best performance for the up-regulated genes, but overall, the differentiation between classes remains minimal.

Our results suggest that our models perform better predicting the host gene expression level during a bacterial infection than the pathogen gene expression level.

4.5 Phylum Assessment

Following model assessment, we investigated to discern whether there are patterns in gene expression prediction that might be tied to broader bacterial evolutionary lineages, by categorizing bacteria at the Phylum level, one of the broader taxonomic classifications, as shown in Table 4.12. The hypothesis is that bacteria within the same Phylum might share certain genomic or regulatory similarities that affect their gene expression profiles. Notably, *Y. ruckeri* underperformed in comparison to other bacteria, despite belonging to the phylum with the highest bacterial count, which

Table 4.12: Bacteria used in our study grouped by Phylum.

Phylum	Bacteria
Bacteroidetes	*Porphyromonas gingivalis*
Proteobacteria	*Yersinia ruckeri* strain YZ
	Haemophilus influenzae Fi176, Hi176
	Bordetella pertussis Tohama I
	Pseudomonas aeruginosa PAO1
Firmicutes	*Streptococcus pyogenes* strain MGAS2221
	Staphylococcus aureus subsp. aureus NCTC 8325
Actinobacteria	*Mycobacterium tuberculosis* H37Rv
	Mycobacterium tuberculosis variant bovis BCG str. ATCC 35733

indicates that the number of species within a given phylum does not necessarily correlate with predictive accuracy for gene expression.

4.6 GO Enrichment Analysis

In our study, we performed GO enrichment analysis on the predicted UP and DOWN genes to investigate whether certain functions were over-represented among the genes predicted by our model.

After classifying the DEGs into ND, UP, and DOWN genes using our models, we assessed the GO enrichment analysis of the predicted UP and DOWN genes using STRING (version 12.0) [87]. STRING has a comprehensive database, encompassing numerous organisms. STRING integrates experimental data, computational predic-

Table 4.13: Analysis of GO enrichment of predicted ND, UP and DOWN genes for host validation datasets

Organism	Term ID	Term Description	Observed Gene count	Background Gene Count	Strength	FDR
Homo sapiens	GO:0006807	Nitrogen compound metabolic process	344	6643	0.1	0.0035
	GO:0044237	Cellular metabolic process	343	6568	0.1	0.0035
	GO:0044238	Primary metabolic process	370	7156	0.09	0.0035
	GO:0071704	Organic substance metabolic process	384	7522	0.09	0.0035
	GO:0008152	Metabolic process	399	7988	0.08	0.0063
	GO:0034641	Cellular nitrogen compound metabolic process	198	3463	0.14	0.0063
	GO:0043170	Macromolecule metabolic process	302	5781	0.1	0.0077
	GO:0006139	Nucleobase-containing compound metabolic process	158	2722	0.14	0.0271
	GO:0046483	Heterocycle metabolic process	165	2891	0.14	0.0361
Macaca fascicularis	GO:0046777	Protein autophosphorylation	29	183	0.51	0.0033
	GO:0001819	Positive regulation of cytokine production	49	466	0.33	0.0170
	GO:0006468	Protein phosphorylation	67	742	0.26	0.0241
	GO:0009987	Cellular process	866	16285	0.03	0.0305
	GO:0016310	Phosphorylation	82	1003	0.22	0.0424
Ictalurus punctatus	GO:0005622	Intracellular anatomical structure	960	16117	0.03	0.0084
	GO:0005737	Cytoplasm	750	12322	0.04	0.0207
	GO:0043227	Membrane-bounded organelle	743	12169	0.05	0.0207
	GO:0005746	Mitochondrial respirasome	14	73	0.54	0.0297
	GO:0043226	Organelle	825	13770	0.04	0.0297
	GO:0043231	Intracellular membrane-bounded organelle	705	11569	0.04	0.0297
	GO:0070469	Respirasome	16	93	0.5	0.0297
	GO:0098803	Respiratory chain complex	15	80	0.53	0.0297

tion methods, and public text collections [87].

For our analysis, we first predicted the gene expression levels in host and pathogen test dataset. Genes were subsequently sorted based on the probability of their classification into one of the three classes. We then selected the top 30% predicted genes from the host organisms and the top 40% from the pathogen. The reasons behind these specific percentages were:

1. to ensure that the chosen genes were not merely random predictions,

2. to maximize the number of enriched terms derived from our analysis.

We did not include *M. tuberculosis* in the STRING pathogen dataset, since the annotation file did not have gene symbols to use in STRING.

Tables 4.13 and 4.14 show the GO terms found to be over-represented among the top UP and DOWN predicted genes. In these tables, the data are presented in four columns. The *Observed Gene Count* details the number of genes in our specific network that are annotated with a given GO term. The *Background Gene Count* provides the total number of genes, encompassing those within our network and in the background dataset, annotated with the same term. The *Strength* column is expressed as $\log_{10}\left(\frac{\text{observed}}{\text{expected}}\right)$, quantifying the enrichment effect. It reflects the ratio of the number of genes in our network annotated with a term to the expected count of such annotations in a random network of the same size. This metric indicates how much more frequently certain terms are annotated in our network compared to what might be expected by chance. Finally, the *FDR* column provides a statistical

Table 4.14: Analysis of GO enrichment of predicted ND, UP and DOWN genes for bacteria validation datasets

Organism	Term ID	Term Description	Observed Gene count	Background Gene Count	Strength	FDR
Y. ruckeri	GO:0008152	Metabolic process	114	1605	0.14	0.00093
	GO:0071704	Organic substance metabolic process	103	1423	0.15	0.0022
	GO:0006807	Nitrogen compound metabolic process	84	1100	0.17	0.0047
	GO:0009987	Cellular process	135	2143	0.09	0.0047
	GO:0044237	Cellular metabolic process	98	1358	0.15	0.0047
	GO:0044238	Primary metabolic process	90	1210	0.16	0.0047
	GO:1901566	Organonitrogen compound biosynthetic process	44	436	0.29	0.0047
	GO:0034641	Cellular nitrogen compound metabolic process	59	717	0.2	0.0164
	GO:0044249	Cellular biosynthetic process	55	658	0.21	0.0164
	GO:1901564	Organonitrogen compound metabolic process	60	733	0.2	0.0164
	GO:1901576	Organic substance biosynthetic process	56	669	0.21	0.0164
	GO:0010467	Gene expression	28	251	0.34	0.0177
	GO:0006412	Translation	16	113	0.44	0.0478
S. pyogenes	GO:1901564	Organonitrogen compound metabolic process	34	440	0.29	0.0269
	GO:0006807	Nitrogen compound metabolic process	45	718	0.2	0.0462
	GO:0044271	Cellular nitrogen compound biosynthetic process	23	260	0.35	0.0462
	GO:0071704	Organic substance metabolic process	51	884	0.16	0.0462
	GO:1901566	Organonitrogen compound biosynthetic process	23	257	0.35	0.0462

significance measure for the observed enrichments. These values are p-values adjusted using the Benjamini–Hochberg procedure [88] to correct for multiple testing within each category. The FDR offers a critical balance between identifying as many relevant terms as possible and limiting the inclusion of terms that might appear significant only due to random variation in the data.

In this analysis, there are GO terms that are associated with infection. For example, Nitrogen-related GO terms (e.g., GO:0006807, GO:1901566, and GO:0034641) as Nitrogen availability affects the progression of a bacterial infection [89, 90]. Posi-

tive regulation of cytokine production (GO:0001819) as cytokines are involved in the host defense against pathogens [91]. Phosphorylation (GO:0006468, GO:0046777 and GO:0016310) has also been associated with viral and bacterial infections [92, 93, 94]. Heterocyclic compounds (GO:0046483) also play a role during infection [95, 96]. Finally, the mitochondria (GO:0005746 and GO:0070469) also play a role in fighting infections [97]. These results suggest that the top predicted genes are indeed involved in infection-related processes, and even though our classifiers show a moderate performance in terms of AUROC, the GO analysis suggests a reasonable performance in identifying DEGs that play a role in bacterial infection.

4.7 Summary

In this chapter, we presented and discussed the results from our study predicting host and pathogen up and down-regulated genes during infection.

We assessed 45 machine learning models generated by combining dimensionality reduction and classification algorithms. For the host, the top models were XGBoost + mRMR (500 features), Random Forest + mRMR (400 features), and LightGBM + mRMR (600 features). For the pathogen, the leading models were XGBoost + PCA (first 30 components), Random Forest + PCA (first 20 components), and LightGBM + mRMR (600 features). After hyperparameter optimization, the selected models were Random Forest + mRMR (400 features) for host and Random Forest + PCA (first 20 components) for pathogen.

We then analyzed feature importance to understand how the selected features contribute to model predictions and demonstrated removing features with negative permutation importance scores did not improve performance, indicating complex feature relationships.

Following feature analysis, the models were trained on training data and the performance was evaluated using 10-fold cross-validation. The host and pathogen models achieved consistent performance across folds for all classes, with a macro-average AUROC score of $71.06\% \pm 1.82\%$ and $66.14\% \pm 0.73\%$, respectively. The models were then tested on independent validation data. The host model performed better than random on human and channel catfish test data with a macro-average AUROC score of 0.60 and 0.57, respectively. The pathogen model also performed slightly better than random for *M. tuberculosis* and *S. pyogenes* test data with a macro-average AUROC score of 0.55 and 0.53, respectively.

Finally, we performed a GO enrichment analysis and found several GO terms related to infection over-represented among the top predicted genes.

Chapter 5

Conclusion

In this study, we developed a dual RNA-seq bioinformatics pipeline utilizing accurate and fast bioinformatics tools. Additionally, for the first time, we used machine learning for predicting gene expression levels during an infection using dual RNA-seq data to label genes for training the classifiers. We explored dimensionality reduction techniques (PCA, VAE, mRMR) and machine learning algorithms (Random Forest, XGBoost, LightGBM), implementing two machine learning models that achieved above-random performance in predicting expression levels for host and pathogen during a bacterial infection.

This thesis contributes to the field of bioinformatics as follows:

- **Development of Dual RNA-seq Bioinformatics Pipeline:** A dual RNA-seq bioinformatics pipeline was developed which is available at https://github.com/BioinformaticsLabAtMUN/DualRNA-infection.

- **Demonstrating the feasibility of learning task:** For the first time, we demonstrated that it is feasible to generate a model for predicting gene expression levels during infection from RNA sequences. The models exhibited a macro-average AUROC scores of 71.06%±1.82% and 66.14%±0.73% for the host and pathogen classifiers, respectively in a 10-fold cross-validation on training set.

A limitation is the limited dataset size and the model's generalizability across different pathogen species. Expanding the dataset size and exploring models with broader generalizability could be steps forward.

For future work, deeper exploration into alternative dimensionality reduction techniques, machine learning methods, and sequence encodings is proposed. This is anticipated to yield novel insights from dual RNA-seq experiments.

In conclusion, this thesis establishes a framework for utilizing machine learning to derive novel insights from dual RNA-seq experiments, laying a foundation for further advancements in this field.

Bibliography

[1] Alexander J. Westermann, Stanislaw A. Gorski, and Jörg Vogel. Dual RNA-seq of pathogen and host. *Nature Reviews Microbiology*, 10(9):618–630, 2012. doi: 10.1038/nrmicro2852. URL www.doi.org/10.1038/nrmicro2852.

[2] James W Marsh, Regan J Hayward, Amol C Shetty, Anup Mahurkar, Michael S Humphrys, and Garry S A Myers. Bioinformatic analysis of bacteria and host cell dual RNA-sequencing experiments. *Briefings in Bioinformatics*, 19(6):1115–1129, 2017. ISSN 1477-4054. doi: 10.1093/bib/bbx043. URL https://doi.org/10.1093/bib/bbx043.

[3] Rotem Sorek and Pascale Cossart. Prokaryotic transcriptomics: a new view on regulation, physiology and pathogenicity. *Nature Reviews Genetics*, 11(1):9–16, 2010. doi: 10.1038/nrg2695. URL https://doi.org/10.1038/nrg2695.

[4] Alexander J. Westermann, Konrad U. Förstner, Fabian Amman, et al. Dual RNA-seq unveils noncoding RNA functions in host–pathogen interactions. *Na-*

ture, 529(7587):496–501, 2016. doi: 10.1038/nature16547. URL www.doi.org/10.1038/nature16547.

[5] Eliza JR Peterson, Rebeca Bailo, Alissa C Rothchild, Mario L Arrieta-Ortiz, Amardeep Kaur, Min Pan, Dat Mai, Abrar A Abidi, Charlotte Cooper, Alan Aderem, Apoorva Bhatt, and Nitin S Baliga. Path-seq identifies an essential mycolate remodeling program for mycobacterial host adaptation. *Molecular Systems Biology*, 15(3), mar 2019. doi: 10.15252/msb.20188584. URL https://doi.org/10.15252%2Fmsb.20188584.

[6] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*, 321:321–331, 2018. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2018.09.013. URL https://www.sciencedirect.com/science/article/pii/S0925231218310749.

[7] Mogana Darshini Ganggayah, Nur Aishah Taib, Yip Cheng Har, Pietro Lio, and Sarinder Kaur Dhillon. Predicting factors for survival of breast cancer patients using machine learning techniques. *BMC Medical Informatics and Decision Making*, 19(1), 2019. doi: 10.1186/s12911-019-0801-4. URL https://doi.org/10.1186/s12911-019-0801-4.

[8] Ahmet Sureyya Rifaioglu, Tunca Doğan, Maria Jesus Martin, Rengul Cetin-

Atalay, and Volkan Atalay. DEEPred: Automated Protein Function Prediction with Multi-task Feed-forward Deep Neural Networks. *Scientific Reports*, 9 (1), 2019. doi: 10.1038/s41598-019-43708-3. URL https://doi.org/10.1038%2Fs41598-019-43708-3.

[9] Likai Wang, Yanpeng Xi, Sibum Sung, et al. RNA-seq assistant: machine learning based methods to identify more transcriptional regulated genes. *BMC Genomics*, 19(1), 2018. doi: 10.1186/s12864-018-4932-2. URL www.doi.org/10.1186/s12864-018-4932-2.

[10] Geng Chen, Baitang Ning, and Tieliu Shi. Single-Cell RNA-seq Technologies and Related Computational Data Analysis. *Frontiers in Genetics*, 10, 2019. doi: 10.3389/fgene.2019.00317. URL www.doi.org/10.3389%2Ffgene.2019.00317.

[11] Raphael Petegrosso, Zhuliu Li, and Rui Kuang. Machine learning and statistical methods for clustering single-cell RNA-sequencing data. *Briefings in Bioinformatics*, 21(4):1209–1223, 2019. doi: 10.1093/bib/bbz063. URL www.doi.org/10.1093/bib/bbz063.

[12] Muta Tah Hira, M. A. Razzaque, Claudio Angione, et al. Integrated multi-omics analysis of ovarian cancer using variational autoencoders. *Scientific Reports*, 11(1), 2021. doi: 10.1038/s41598-021-85285-4. URL www.doi.org/10.1038/s41598-021-85285-4.

[13] Adonis D'Mello, Ashleigh N. Riegler, Eriel Martínez, et al. An in vivo atlas

of host–pathogen transcriptomes during *Streptococcus pneumoniae* colonization and disease. *Proceedings of the National Academy of Sciences*, 117(52):33507–33518, 2020. doi: 10.1073/pnas.2010428117. URL www.doi.org/10.1073/pnas.2010428117.

[14] Alexander J. Westermann, Lars Barquist, and Jörg Vogel. Resolving host–pathogen interactions by dual RNA-seq. *PLOS Pathogens*, 13(2):e1006033, 2017. doi: 10.1371/journal.ppat.1006033. URL www.doi.org/10.1371/journal.ppat.1006033.

[15] Regan J. Hayward, Michael S. Humphrys, Wilhelmina M. Huston, et al. Dual RNA-seq analysis of in vitro infection multiplicity and RNA depletion methods in Chlamydia-infected epithelial cells. *Scientific Reports*, 11(1), 2021. doi: 10.1038/s41598-021-89921-x. URL www.doi.org/10.1038/s41598-021-89921-x.

[16] Steve Minchin and Julia Lodge. Understanding biochemistry: structure and function of nucleic acids. *Essays in Biochemistry*, 63(4):433–456, 10 2019. ISSN 0071-1365. doi: 10.1042/EBC20180038. URL https://doi.org/10.1042/EBC20180038.

[17] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, Peter Walter, et al. *Molecular Biology of the Cell*. Garland Science, New York, 4th edition, 2002. https://www.ncbi.nlm.nih.gov/books/NBK26887/.

[18] Robert G. Nichols and Emily R. Davenport. The relationship between the gut

microbiome and host gene expression: a review. *Human Genetics*, 140(5):747–760, 2021. doi: 10.1007/s00439-020-02237-0. URL https://doi.org/10.1007/s00439-020-02237-0.

[19] Anthony M. Bolger, Marc Lohse, and Bjoern Usadel. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, 30(15):2114–2120, 04 2014. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu170. URL https://doi.org/10.1093/bioinformatics/btu170.

[20] Andrews, Simon. FastQC. https://www.bioinformatics.babraham.ac.uk/projects/fastqc/, 2010. [Software].

[21] Marcel Martin. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal*, 17(1):10–12, May 2011. doi: http://dx.doi.org/10.14806/ej.17.1.200. URL http://journal.embnet.org/index.php/embnetjournal/article/view/200.

[22] Leng Wu, Rui Shi, Huimin Bai, Xingtong Wang, Jian Wei, Chengcheng Liu, and Yafei Wu. *Porphyromonas gingivalis* induces increases in branched-chain amino acid levels and exacerbates liver injury through livh/livk. *Frontiers in Cellular and Infection Microbiology*, 12, mar 2022. doi: 10.3389/fcimb.2022.776996. URL https://doi.org/10.3389%2Ffcimb.2022.776996.

[23] Yibin Yang, Xia Zhu, Haixin Zhang, Yuhua Chen, Yi Song, and Xiaohui Ai. Dual RNA-seq of trunk kidneys extracted from channel catfish infected with

Yersinia ruckeri reveals novel insights into host-pathogen interactions. *Frontiers in Immunology*, 12, dec 2021. doi: 10.3389/fimmu.2021.775708. URL https://doi.org/10.3389%2Ffimmu.2021.775708.

[24] Priyanka Kachroo, Jesus M. Eraso, Randall J. Olsen, Luchang Zhu, Samantha L. Kubiak, Layne Pruitt, Prasanti Yerramilli, Concepcion C. Cantu, Matthew Ojeda Saavedra, Johan Pensar, Jukka Corander, Leslie Jenkins, Lillian Kao, Alejandro Granillo, Adeline R. Porter, Frank R. DeLeo, and James M. Musser. New pathogenesis mechanisms and translational leads identified by multidimensional analysis of necrotizing myositis in primates. *mBio*, 11(1), feb 2020. doi: 10.1128/mbio.03363-19. URL https://doi.org/10.1128%2Fmbio.03363-19.

[25] Mariam R. Farman, Denisa Petráčková, Dilip Kumar, Jakub Držmíšek, Argha Saha, Ivana Čurnová, Jan Čapek, Václava Hejnarová, Fabian Amman, Ivo Hofacker, and Branislav Večerek. Avirulent phenotype promotes *Bordetella pertussis* adaptation to the intramacrophage environment. *Emerging Microbes & Infections*, 12(1):e2146536, 2023. doi: 10.1080/22221751.2022.2146536. URL https://doi.org/10.1080/22221751.2022.2146536. PMID: 36357372.

[26] Oliver Goldmann, Till Sauerwein, Gabriella Molinari, Manfred Rohde, Konrad U. Förstner, and Eva Medina. Cytosolic Sensing of Intracellular *Staphylococcus aureus* by Mast Cells Elicits a Type I IFN Response That Enhances

Cell-Autonomous Immunity. *The Journal of Immunology*, 208(7):1675–1685, 04 2022. ISSN 0022-1767. doi: 10.4049/jimmunol.2100622. URL https://doi.org/10.4049/jimmunol.2100622.

[27] Alexander Dobin, Carrie A. Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R. Gingeras. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21, 10 2012. ISSN 1367-4803. doi: 10.1093/bioinformatics/bts635. URL https://doi.org/10.1093/bioinformatics/bts635.

[28] Cole Trapnell, Lior Pachter, and Steven L Salzberg. Tophat: discovering splice junctions with RNA-seq. *Bioinformatics*, 25(9):1105–1111, May 2009. ISSN 1367-4811 (Electronic); 1367-4803 (Print); 1367-4803 (Linking). doi: 10.1093/bioinformatics/btp120.

[29] Tanja Magoc, Derrick Wood, and Steven L Salzberg. Edge-pro: Estimated degree of gene expression in prokaryotic genomes. *Evol Bioinform Online*, 9:127–136, 2013. ISSN 1176-9343 (Print); 1176-9343 (Electronic); 1176-9343 (Linking). doi: 10.4137/EBO.S11250.

[30] Ben Langmead and Steven L Salzberg. Fast gapped-read alignment with bowtie 2. *Nature Methods*, 9(4):357–359, 2012. doi: 10.1038/nmeth.1923. URL https://doi.org/10.1038/nmeth.1923.

[31] Steve Hoffmann, Christian Otto, Stefan Kurtz, Cynthia M. Sharma, Philipp

Khaitovich, Jörg Vogel, Peter F. Stadler, and Jörg Hackermüller. Fast mapping of short sequences with mismatches, insertions and deletions using index structures. *PLOS Computational Biology*, 5(9):1–10, 09 2009. doi: 10.1371/journal.pcbi.1000502. URL https://doi.org/10.1371/journal.pcbi.1000502.

[32] F. Heath Damron, Amanda G. Oglesby-Sherrouse, Angela Wilks, and Mariette Barbier. Dual-seq transcriptomics reveals the battle for iron during *Pseudomonas aeruginosa* acute murine pneumonia. *Scientific Reports*, 6(1), dec 2016. doi: 10.1038/srep39172. URL https://doi.org/10.1038%2Fsrep39172.

[33] Yang Liao, Gordon K. Smyth, and Wei Shi. featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics*, 30(7):923–930, 11 2013. ISSN 1367-4803. doi: 10.1093/bioinformatics/btt656. URL https://doi.org/10.1093/bioinformatics/btt656.

[34] Bo Li and Colin N. Dewey. Rsem: accurate transcript quantification from RNA-seq data with or without a reference genome. *BMC Bioinformatics*, 12(1): 323, 2011. doi: 10.1186/1471-2105-12-323. URL https://doi.org/10.1186/1471-2105-12-323.

[35] Buket Baddal, Alessandro Muzzi, Stefano Censini, Raffaele A. Calogero, Giulia Torricelli, Silvia Guidotti, Anna R. Taddei, Antonello Covacci, Mariagrazia Pizza, Rino Rappuoli, Marco Soriani, and Alfredo Pezzicoli. Dual RNA-seq of nontypeable *Haemophilus influenzae* and host cell transcriptomes reveals

novel insights into host-pathogen cross talk. *mBio*, 6(6), dec 2015. doi: 10.1128/mbio.01765-15. URL https://doi.org/10.1128%2Fmbio.01765-15.

[36] Simon Anders, Paul Theodor Pyl, and Wolfgang Huber. HTSeq—a Python framework to work with high-throughput sequencing data. *Bioinformatics*, 31 (2):166–169, 09 2014. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu638. URL https://doi.org/10.1093/bioinformatics/btu638.

[37] Rob Patro, Geet Duggal, Michael I Love, Rafael A Irizarry, and Carl Kingsford. Salmon provides fast and bias-aware quantification of transcript expression. *Nat Methods*, 14(4):417–419, Apr 2017. ISSN 1548-7105 (Electronic); 1548-7091 (Print); 1548-7091 (Linking). doi: 10.1038/nmeth.4197.

[38] Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Nature Precedings*, mar 2010. doi: 10.1038/npre.2010.4282.1. URL https://doi.org/10.1038%2Fnpre.2010.4282.1.

[39] Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with deseq2. *Genome biology*, 15 (12):1–21, 2014. doi: 10.1186/s13059-014-0550-8.

[40] Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, Jan 2010. ISSN 1367-4811 (Electronic); 1367-4803 (Print); 1367-4803 (Linking). doi: 10.1093/bioinformatics/btp616.

[41] Matthew E Ritchie, Belinda Phipson, Di Wu, Yifang Hu, Charity W Law, Wei Shi, and Gordon K Smyth. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res*, 43(7):e47, Apr 2015. ISSN 1362-4962 (Electronic); 0305-1048 (Print); 0305-1048 (Linking). doi: 10.1093/nar/gkv007.

[42] Davide Risso, John Ngai, Terence P Speed, and Sandrine Dudoit. Normalization of RNA-seq data using factor analysis of control genes or samples. *Nature Biotechnology*, 32(9):896–902, 2014. doi: 10.1038/nbt.2931. URL https://doi.org/10.1038/nbt.2931.

[43] K.S.Mehta, D.S.Mehta, and V.Dahiya. A comparative study of computational tools for biological sequence cleaning and analysis. *International Journal of Computer Sciences and Engineering*, 6:1136–1140, 7 2018. ISSN 2347-2693. doi: https://doi.org/10.26438/ijcse/v6i7.11361140. URL https://www.ijcseonline.org/full_paper_view.php?paper_id=2573.

[44] Dongmei Li, Martin S. Zand, Timothy D. Dye, Maciej L. Goniewicz, Irfan Rahman, and Zidian Xie. An evaluation of RNA-seq differential analysis methods. *PLOS ONE*, 17(9):1–19, 09 2022. doi: 10.1371/journal.pone.0264246. URL https://doi.org/10.1371/journal.pone.0264246.

[45] Stephanie Schaarschmidt, Axel Fischer, Ellen Zuther, and Dirk K. Hincha. Evaluation of seven different RNA-seq alignment tools based on experimental data

from the model plant *Arabidopsis* thaliana. *International Journal of Molecular Sciences*, 21(5):1720, Mar 2020. ISSN 1422-0067. doi: 10.3390/ijms21051720. URL http://dx.doi.org/10.3390/ijms21051720.

[46] Tulika Kakati, Dhruba K. Bhattacharyya, Jugal K. Kalita, and Trina M. Norden-Krichmar. DEGnext: classification of differentially expressed genes from RNA-seq data using a convolutional neural network with transfer learning. *BMC Bioinformatics*, 23(1):17, 2022. doi: 10.1186/s12859-021-04527-4. URL https://doi.org/10.1186/s12859-021-04527-4.

[47] Kyle Chang, Chad J Creighton, Caleb Davis, et al. The cancer genome atlas pan-cancer analysis project. *Nature Genetics*, 45(10):1113–1120, 2013. doi: 10.1038/ng.2764. URL https://doi.org/10.1038/ng.2764.

[48] Wei Li, Yanbin Yin, Xiongwen Quan, and Han Zhang. Gene expression value prediction based on XGBoost algorithm. *Frontiers in Genetics*, 10, 2019. ISSN 1664-8021. doi: 10.3389/fgene.2019.01077. URL https://www.frontiersin.org/articles/10.3389/fgene.2019.01077.

[49] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

[50] Tanya Barrett, Stephen E. Wilhite, Pierre Ledoux, Carlos Evangelista, Irene F. Kim, Maxim Tomashevsky, Kimberly A. Marshall, Katherine H. Phillippy, Patti M. Sherman, Michelle Holko, Andrey Yefanov, Hyeseung Lee, Naigong Zhang, Cynthia L. Robertson, Nadezhda Serova, Sean Davis, and Alexandra Soboleva. NCBI GEO: archive for functional genomics data sets—update. *Nucleic Acids Research*, 41(D1):D991–D995, 11 2012. ISSN 0305-1048. doi: 10.1093/nar/gks1193. URL https://doi.org/10.1093/nar/gks1193.

[51] The GTEx Consortium et al. The genotype-tissue expression (gtex) pilot analysis: Multitissue gene regulation in humans. *Science*, 348(6235):648–660, 2015. doi: 10.1126/science.1262110. URL https://www.science.org/doi/abs/10.1126/science.1262110.

[52] Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R. Ledsam, Agnieszka Grabska-Barwinska, Kyle R. Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R. Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods*, 18(10):1196–1203, oct 2021. doi: 10.1038/s41592-021-01252-x. URL https://doi.org/10.1038%2Fs41592-021-01252-x.

[53] Eric W Sayers, Evan E Bolton, J Rodney Brister, Kathi Canese, Jessica Chan, Donald C Comeau, Ryan Connor, Kathryn Funk, Chris Kelly, Sunghwan Kim, Tom Madej, Aron Marchler-Bauer, Christopher Lanczycki, Stacy

Lathrop, Zhiyong Lu, Francoise Thibaud-Nissen, Terence Murphy, Lon Phan, Yuri Skripchenko, Tony Tse, Jiyao Wang, Rebecca Williams, Barton W Trawick, Kim D Pruitt, and Stephen T Sherry. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 50(D1): D20–D26, 12 2021. ISSN 0305-1048. doi: 10.1093/nar/gkab1112. URL https://doi.org/10.1093/nar/gkab1112.

[54] Nuala A. O'Leary, Mathew W. Wright, J. Rodney Brister, et al. Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Research*, 44(D1):D733–D745, 11 2015. ISSN 0305-1048. doi: 10.1093/nar/gkv1189. URL https://doi.org/10.1093/nar/gkv1189.

[55] Rienk A Rienksma, Maria Suarez-Diez, Hans-Joachim Mollenkopf, Gregory M Dolganov, Anca Dorhoi, Gary K Schoolnik, Vitor AP Martins dos Santos, Stefan HE Kaufmann, Peter J Schaap, and Martin Gengenbacher. Comprehensive insights into transcriptional adaptation of intracellular mycobacteria by microbe-enriched dual RNA sequencing. *BMC Genomics*, 16(1), feb 2015. doi: 10.1186/s12864-014-1197-2. URL https://doi.org/10.1186%2Fs12864-014-1197-2.

[56] National Center for Biotechnology Information (NCBI). NCBI Datasets Command Line Tools. https://www.ncbi.nlm.nih.gov/datasets/docs/command-line-tools/, 2021. [Software].

[57] Rasko Leinonen, Hideaki Sugawara, Martin Shumway, and on behalf of the International Nucleotide Sequence Database Collaboration. The sequence read archive. *Nucleic Acids Research*, 39(suppl_1):D19–D21, 10/13/2023 2011. doi: 10.1093/nar/gkq1019. URL https://doi.org/10.1093/nar/gkq1019.

[58] Rasko Leinonen, Ruth Akhtar, Ewan Birney, Lawrence Bower, Ana Cerdeno-Tárraga, et al. The European Nucleotide Archive. *Nucleic Acids Research*, 39, 10 2010. ISSN 0305-1048. doi: 10.1093/nar/gkq967. URL https://doi.org/10.1093/nar/gkq967.

[59] National Center for Biotechnology Information (NCBI). SRA-Toolkit. https://github.com/ncbi/sra-tools, 2008. [Software].

[60] Shifu Chen, Yanqing Zhou, Yaru Chen, and Jia Gu. fastp: an ultra-fast all-in-one FASTQ preprocessor. *Bioinformatics*, 34(17):i884–i890, 09 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty560. URL https://doi.org/10.1093/bioinformatics/bty560.

[61] Shifu Chen. Ultrafast one-pass fastq data preprocessing, quality control, and deduplication using fastp. *iMeta*, 2(2):e107, 2023. doi: https://doi.org/10.1002/imt2.107. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/imt2.107.

[62] Zhen Yao, Frank M. You, Amidou N'Diaye, Ron E. Knox, Curt McCartney, Colin W. Hiebert, Curtis Pozniak, and Wayne Xu. Evaluation of variant call-

ing tools for large plant genome re-sequencing. *BMC Bioinformatics*, 21(1): 360, 2020. doi: 10.1186/s12859-020-03704-1. URL https://doi.org/10.1186/s12859-020-03704-1.

[63] Junfeng Xia, Jing Shang, Fei Zhu, Wanwipa Vongsangnak, Yifei Tang, Wenyu Zhang, and Bairong Shen. Evaluation and comparison of multiple aligners for next-generation sequencing data analysis. *BioMed Research International*, 2014: 309650, 2014. doi: 10.1155/2014/309650. URL https://doi.org/10.1155/2014/309650.

[64] Dimitra Sarantopoulou, Thomas G. Brooks, Soumyashant Nayak, Antonijo Mrčela, Nicholas F. Lahens, and Gregory R. Grant. Comparative evaluation of full-length isoform quantification from RNA-seq. *BMC Bioinformatics*, 22(1): 266, 2021. doi: 10.1186/s12859-021-04198-1. URL https://doi.org/10.1186/s12859-021-04198-1.

[65] Alex J Tate, Kristen C Brown, and Taiowa A Montgomery. tiny-count: a counting tool for hierarchical classification and quantification of small RNA-seq reads with single-nucleotide precision. *Bioinformatics Advances*, 3(1): vbad065, 05 2023. ISSN 2635-0041. doi: 10.1093/bioadv/vbad065. URL https://doi.org/10.1093/bioadv/vbad065.

[66] Artem Tarasov, Albert J. Vilella, Edwin Cuppen, Isaac J. Nijman, and Pjotr Prins. Sambamba: fast processing of NGS alignment formats. *Bioinformatics*, 31

(12):2032–2034, 02 2015. ISSN 1367-4803. doi: 10.1093/bioinformatics/btv098. URL https://doi.org/10.1093/bioinformatics/btv098.

[67] Li Wang, Sheng Wang, and Wei Li. RSeQC: quality control of RNA-seq experiments. *Bioinformatics*, 28(16):2184–2185, 2012. doi: 10.1093/bioinformatics/bts356.

[68] Dongmei Li, Martin S. Zand, Timothy D. Dye, Maciej L. Goniewicz, Irfan Rahman, and Zidian Xie. An evaluation of rna-seq differential analysis methods. *PLOS ONE*, 17(9):1–19, 09 2022. doi: 10.1371/journal.pone.0264246. URL https://doi.org/10.1371/journal.pone.0264246.

[69] Tukur Dahiru. P - value, a true test of statistical significance? a cautionary note. *Ann Ib Postgrad Med*, 6(1):21–26, Jun 2008. ISSN 1597-1627 (Print); 1597-1627 (Linking). doi: 10.4314/aipm.v6i1.64038.

[70] Jose D. Perezgonzalez. Fisher, Neyman-Pearson or NHST? A tutorial for teaching data testing. *Frontiers in Psychology*, 6, 2015. ISSN 1664-1078. doi: 10.3389/fpsyg.2015.00223. URL https://www.frontiersin.org/articles/10.3389/fpsyg.2015.00223.

[71] Robson P Bonidia, Douglas S Domingues, Danilo S Sanches, and André C P L F de Carvalho. MathFeature: feature extraction package for DNA, RNA and protein sequences based on mathematical descriptors. *Briefings in Bioin-*

formatics, 11 2021. ISSN 1477-4054. doi: 10.1093/bib/bbab434. URL https://doi.org/10.1093/bib/bbab434. bbab434.

[72] Chin Ka Yin, Shoichi Ishida, and Kei Terayama. Predicting condensate formation of protein and RNA under various environmental conditions. *bioRxiv*, 2023. doi: 10.1101/2023.06.01.543215. URL https://www.biorxiv.org/content/early/2023/06/05/2023.06.01.543215.

[73] Dheeraj Raya, Vincent Peta, Alain Bomgni, Tuyen Du Do, Jawaharraj Kalimuthu, David R. Salem, Venkataramana Gadhamshetty, Etienne Z. Gnimpieba, and Saurabh Sudha Dhiman. Classification of bacterial nanowire proteins using machine learning and feature engineering model. *bioRxiv*, 2023. doi: 10.1101/2023.05.03.539336. URL https://www.biorxiv.org/content/early/2023/05/05/2023.05.03.539336.

[74] Robson Parmezan Bonidia, Lucas Dias Hiera Sampaio, Douglas Silva Domingues, Alexandre Rossi Paschoal, Fabrício Martins Lopes, André Carlos Ponce de Leon Ferreira de Carvalho, and Danilo Sipoli Sanches. Feature extraction approaches for biological sequences: A comparative study of mathematical models. *bioRxiv*, 2020. doi: 10.1101/2020.06.08.140368. URL https://www.biorxiv.org/content/early/2020/08/06/2020.06.08.140368.

[75] H. Hotelling. Analysis of a complex of statistical variables into principal com-

ponents. *Journal of Educational Psychology*, 24(6):417–441, sep 1933. doi: 10.1037/h0071325. URL https://doi.org/10.1037%2Fh0071325.

[76] Max Welling and Diederik P Kingma. Auto-encoding variational bayes. *CoRR*, 2014.

[77] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. In *Computational Systems Bioinformatics. CSB2003. Proceedings of the 2003 IEEE Bioinformatics Conference. CSB2003*, pages 523–528, 2003. doi: 10.1109/CSB.2003.1227396.

[78] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

[79] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[80] Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances*

in Neural Information Processing Systems, volume 35, pages 507–520. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/0378c7692da36807bdec87ab043cdadc-Paper-Datasets_and_Benchmarks.pdf.

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

[82] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[83] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

[84] Qi Wei and Stephen A. Ramsey. Predicting chemotherapy response using a

variational autoencoder approach. *BMC Bioinformatics*, 22(1), 2021. doi: 10.1186/s12859-021-04339-6. URL www.doi.org/10.1186/s12859-021-04339-6.

[85] Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997. ISSN 0031-3203. doi: https://doi.org/10.1016/S0031-3203(96)00142-2. URL https://www.sciencedirect.com/science/article/pii/S0031320396001422.

[86] James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 115–123, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/bergstra13.html.

[87] Damian Szklarczyk, Annika L Gable, Katerina C Nastou, David Lyon, Rebecca Kirsch, Sampo Pyysalo, Nadezhda T Doncheva, Marc Legeay, Tao Fang, Peer Bork, Lars J Jensen, and Christian von Mering. The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic Acids Research*, 49(D1):D605–D612, 11 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa1074. URL https://doi.org/10.1093/nar/gkaa1074.

[88] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A

practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995. ISSN 00359246. URL http://www.jstor.org/stable/2346101.

[89] Yufan Xu, Shiwei Ma, Zixin Huang, Longlong Wang, Sayed Haidar Abbas Raza, and Zhe Wang. Nitrogen metabolism in mycobacteria: the key genes and targeted antimicrobials. *Frontiers in Microbiology*, 14, 2023. ISSN 1664-302X. doi: 10.3389/fmicb.2023.1149041. URL https://www.frontiersin.org/articles/10.3389/fmicb.2023.1149041.

[90] Mathilde Fagard, Alban Launay, Gilles Clément, Julia Courtial, Alia Dellagi, Mahsa Farjad, Anne Krapp, Marie-Christine Soulié, and Céline Masclaux-Daubresse. Nitrogen metabolism meets phytopathology. *Journal of Experimental Botany*, 65(19):5643–5656, 07 2014. ISSN 0022-0957. doi: 10.1093/jxb/eru323. URL https://doi.org/10.1093/jxb/eru323.

[91] Judith A. Smith. Regulation of cytokine production by the unfolded protein response; implications for infection and autoimmunity. *Frontiers in Immunology*, 9, 2018. ISSN 1664-3224. doi: 10.3389/fimmu.2018.00422. URL https://www.frontiersin.org/articles/10.3389/fimmu.2018.00422.

[92] Alberto Valdés, Hongxing Zhao, Ulf Pettersson, and Sara Bergström Lind. Phosphorylation time-course study of the response during adenovirus type 2 infection. *PROTEOMICS*, 20(7):1900327, 2020. doi: https://doi.org/10.1002/pmic.

201900327. URL https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/pmic.201900327.

[93] W. C. Russell and G. E. Blair. Polypeptide phosphorylation in adenovirus-infected cells. *Journal of General Virology*, 34(1):19–35, 1977. ISSN 1465-2099. doi: https://doi.org/10.1099/0022-1317-34-1-19. URL https://www.microbiologyresearch.org/content/journal/jgv/10.1099/0022-1317-34-1-19.

[94] Julie Bonne Køhler, Carsten Jers, Mériem Senissar, Lei Shi, Abderahmane Derouiche, and Ivan Mijakovic. Importance of protein Ser/Thr/Tyr phosphorylation for bacterial pathogenesis. *FEBS Letters*, 594(15):2339–2369, 2020. doi: https://doi.org/10.1002/1873-3468.13797. URL https://febs.onlinelibrary.wiley.com/doi/abs/10.1002/1873-3468.13797.

[95] Rehab H. Abd El-Aleam, Riham F. George, Hanan H. Georgey, and Hamdy M. Abdel-Rahman. Bacterial virulence factors: a target for heterocyclic compounds to combat bacterial resistance. *RSC Adv.*, 11:36459–36482, 2021. doi: 10.1039/D1RA06238G. URL http://dx.doi.org/10.1039/D1RA06238G.

[96] Andreacarola Urso and Alice Prince. Anti-inflammatory metabolites in the pathogenesis of bacterial infection. *Frontiers in Cellular and Infection Microbiology*, 12, 2022. ISSN 2235-2988. doi: 10.3389/fcimb.2022.925746. URL https://www.frontiersin.org/articles/10.3389/fcimb.2022.925746.

[97] Pedro Escoll, Lucien Platon, and Carmen Buchrieser. Roles of mitochondrial respiratory complexes during infection. *Immunometabolism*, 1(2), 2019. URL https://journals.lww.com/immunometabolism/fulltext/2019/10000/roles_of_mitochondrial_respiratory_complexes.2.aspx.