Unveiling *E. coli* Adaptation Dynamics with protein language model and random walk

A Project Report

submitted by

Nancy Jaiswal

MT22201

in partial fulfillment of the requirements for the award of the degree of

MASTER OF TECHNOLOGY

Department of Computational Biology

INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI NEW DELHI- 110020

May 21, 2024

THESIS CERTIFICATE

This is to certify that the thesis titled Unveiling *E. coli* Adaptation Dynamics with Protein Language Model and Random Walk, submitted by Nancy Jaiswal, to the Indraprastha Institute of Information Technology, New Delhi, for the award of the degree of M. Tech. Computational Biology, is a bona fide record of the research work done by her under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Debarkar Sengupta

Thesis Supervisor

Associate Professor

Dept. of Computational Biology IIIT Delhi, 110020

Place: New Delhi

Date: 21 May 2024

Acknowledgment

I would like to express my heartfelt gratitude to Dr. Debarkar Sengupta for his invaluable support and guidance throughout this research endeavor. Dr. Sengupta's expertise and insights have played a pivotal role in shaping the direction and outcomes of this study. His mentorship has been instrumental in navigating the complexities of our research objectives.

I am also deeply grateful to Bernadette Mathew for her critical guidance, unwavering support, and thoughtful feedback. Her expertise and attention to detail have greatly enriched the quality of our work. Bernadette's contributions have been instrumental in refining our methodologies and interpreting our findings with clarity and precision.

I extend my sincere appreciation to Abhishek Haldar for his valuable advice on this research project.

I would also like to acknowledge the contributions of Harshit Pardhi and Justin Thomas. Their assistance with experimental procedures and dedication to the project have been indispensable to its success.

I am deeply grateful for the collaborative efforts and dedication of all those mentioned above. This research would not have been possible without their collective expertise, support, and commitment to advancing scientific knowledge in our field.

Contents

List of all images

List of all script images

List of all Tables

Abbreviations

LLM - Large Language Model
LLR - Log Likelihood ratio
PLM - Protein Language model
RWR - Random walk Restart
GO - Gene Ontology
E. coli - *Escherichia coli*
VCF - Variant calling file
INT - INT (iodonitrotetrazolium or 2-(4-iodophenyl)-3-(4-nitrophenyl)-5-phenyl-2H-tetrazolium)
PMS - Phenazine methosulphate
ESM - Evolutionary Scale Modeling
LB - Luria Broth
MIC - Minimum Inhibitory Concentration

Abstract

Biological processes rely on intricate interactions among multiple genes, forming diverse networks like protein-protein interaction networks, gene regulation networks, gene co-expression networks, and metabolic networks. Graph theory algorithms analyze these networks to unveil complex biological interactions. The Random Walk with Restart (RWR) algorithm is cutting-edge, extending to multiplex and heterogeneous networks, exploring various layers of gene and protein interactions, including protein-protein interactions and co-expression correlations. Additionally, it transitions to networks reflecting phenotype similarities among genes. We developed a method to decipher the intrinsic phenotype mechanisms through genomic data utilizing RWR-M to identify functionally associated genes, integrated with pathway analysis, facilitating the subsequent identification of pivotal pathways.

To validate our method, we conducted a single-cell bottleneck experiment. We grew the wild-type *E. coli* strain MG1655 under two varying conditions, subjecting them to increasing sublethal antibiotic pressure. Following that, we conducted whole-genome sequencing at every time point for both growth conditions. The variants were identified and they were used as seed genes in RWR-M. The RWR-M network was constructed using STRING, *E. coli* net, and Weighted Gene Co-expression Network Analysis (WGCNA). We subsequently analyzed the gene set through pathway analysis to pinpoint the crucial pathways and their associated genes responsible for the observed phenotypic variances in the two distinct environmental conditions. We supported our finding using protein language model (PLMs) and literature survey.

By employing a multifaceted approach that combines various methodologies, we have established a comprehensive framework for pinpointing the critical factors responsible for the observed phenotypic changes.

Chapter 1

Introduction

How do living cells work as vibrant and complex ecosystems full of biomolecular interactions? These interactions cover protein-protein binding, metabolic exchanges, and gene regulation in the cellular interactome, an integrated, sophisticated network fundamental to life[1]. Knowing the interactome is one of the major challenges in modern biology for human health, disease, and evolution[2].

One such revolutionary approach is network biology, which shifted the focus from individual molecules to the cellular network analysis of biomolecules. Therein, the biomolecules – genes, proteins, and metabolites are represented as nodes, and the interactions between these biomolecules are in form of edges. Then, with the use of graph theory and network algorithms, these complex networks can be dissected, and its components key components – hub genes, essential proteins, hidden patterns – co-expressed genes, regulatory pathways prediction of the system behavior in response to various perturbations – environmental changes, drug treatment treated [3].

Network biology's practical applications are deep. In metabolic engineering, network models have played a major role in the design of artificial pathways for production of valuable biomolecules. Diseases affecting humans can also be understood through network analysis, which helps identify network modules that contribute to disease progression and development of targeted therapies as well as novel drug targets [4]. For example, in cancer and neurodegeneration diseases network analysis has been a mainstay for understanding gene-gene interactions.

Nevertheless, the complexity of the interactome is a serious hurdle. These huge networks are highly dynamic and their experimental and computational techniques are currently not adequate. There has been an explosion in high-throughput

technologies such as next generation sequencing which produced huge information on biological interactions [5]. However, turning this data into actionable insights is insurmountable.

The question can be tackled by network biology along with its powerful tools using network algorithms. For integrating, analyzing, and modeling complex biological datasets, these tools provide a strong structure [6]. When researchers use machine learning techniques, statistical inference as well as optimization methods they can identify robust patterns within networks; predict new interactions and design targeted interventions. Furthermore, the development of user-friendly software tools and databases has increased access to network biology for a wider range of researchers [7].

This study investigates how Random Walk with Restart (RWR), a major network algorithm can be used in the analysis of intricate biological systems. This will include looking at:

These are mechanisms that drive network rewiring over time.

These are evolutionary processes that affect organismal fitness and adaptation due to the properties of networks.

Network-based drug discovery: Identifying new therapeutic targets and predicting drug efficacy through network analysis.

Disease pathway elucidation: Using network analysis to unravel the complex pathways underlying human diseases [8].

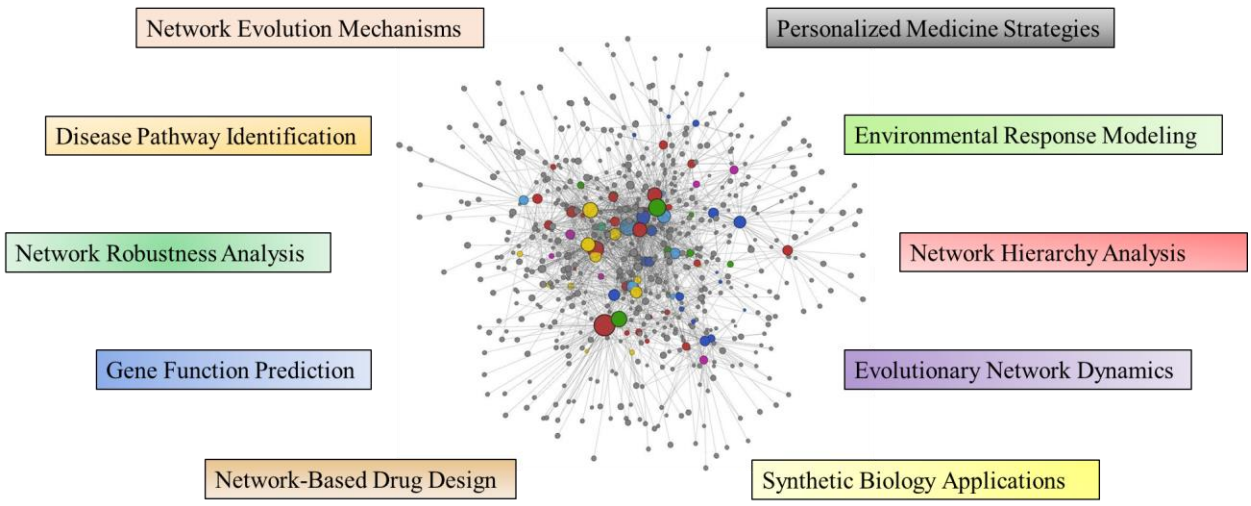


Figure1

https://slideplayer.com/slide/10326065/#google_vignette

Protein language models can improve our understanding of interactions in biological networks.

Through the application of these modern tools, we seek to gain more insights into the molecular language of life represented by the inner workings of cellular interactome. The study of network biology would revolutionize our comprehension of cellular activities. This is because by untangling the intricate interplays that occur within these nets, scientists can not only add on their fundamental biological knowledge but also develop new curative measures hence enhancing the delicacy and complexity within living bodies.

Chapter 2

Objective

The primary objective of this research is to elaborate on the development path of bacteria population under increased stress levels of antibiotics. The D population's main hypothesis argues that they will generate useful mutations due to increasing antibiotic concentrations in rich Luria broth (LB) medium. These mutations will enable them to grow under antibiotic-free and antibiotic-containing conditions. Conversely, the R population grown in selective M9 minimal media may have low viability on exposure to antibiotics because harmful mutations constraining growth regardless of presence or absence of antibiotics tend to accumulate.

We use a network algorithm and a protein language model so as to comprehensively define the mutational landscape of both populations. By applying this approach, we can identify the precise metabolic pathways affected by these mutations and then predict how these might affect downstream protein structure and function. In this diverse study, we seek to classify with a certain amount of certainty if the resulting mutations are pathogenic (worse off bacterial fitness) or benign (no effect on bacterial survival)

Chapter 3

Material and Methods

1) Algorithm

A) Random walk with restart

Random walk with restart (RWR) is a state-of-the-art algorithm that is a part of network biology. It has been vastly utilized for many applications, i.e., key gene prioritization, driver gene prediction integrating phenotypic datasets, finding the relationship between two nodes. Using RWR the network nodes are ranked according to their closeness with the starting nodes i.e., seed node [9]. RWR ensures that the seed walk is not lost in the network or trapped in a dead end and assures the existence of stable distribution. RWR-H (multiplex homogenous network) and RWR-MH (multiplex heterogeneous network) are the two applications in the inbuilt R package for RWR. RWR-H (multiplex homogenous network) integrated by 3 layers where every layer has information of links among genes (PPI, pathways and co-expression) [10].

Random walk with restart works as follows
Initialization:

1) Initially, the networks were represented with an adjacency matrix that has information about connections and weights between nodes.
2) The adjacency matrix is converted to a transition matrix having information about the probability of moving from one node to another
3) An initial probability vector is set in which the starting node is given a high value and zero for all other nodes.

Process:

1) The process starts by performing a random walk at the starting node.
2) Based on probabilities given by the transition matrix the walker moves from one node to another node

3) Based on the restart probability the walker will return to the starting node at each step

This process continues repeatedly. The probability distribution of the walker's position over the nodes in the network is adjusted at every iteration, with probabilities of moving to neighboring nodes being combined with that of returning to the starting node.

Iterations go on unless a stable distribution is achieved where probabilities no longer change much between steps.

So when the stable distribution is achieved, the resulting probability indicates those nodes that are visited more frequently. That means the nodes that are closer to the starting node have higher probability [11] [12] [13].

In this work we used three networks to do multiplexing which is as follows:
STRING: It is a search tool for the retrieval of interacting genes or proteins. This network was downloaded from the string website. We downloaded the PPI network. It builds a network that demonstrates the connection between proteins. The network consists of both direct interaction and functional association.
EcoliNet: We downloaded the EcoliNet v.1 network which comprises of 4099 genes and 95520 links. It is a network produced by integration of all data-type-specific networks CC (co-citation of two genes), CX (co-expression pattern of two genes), DC (co-occurrence of protein domains between two coding genes) GN (similar genomic context of bacterial orthologs of two *E. coli* genes), HT (high-throughput protein-protein interactions), LC (small/medium-scale protein-protein interactions), PG (similar phylogenetic profiles between two E. coli genes).
WGCNA: Weighted Gene Co-expression Network Analysis work on the principle that similar expression patterns of genes are a part of same module. These modules are functionally related meaning that the genes within a module might work together in a biological pathway. We used this method to create a co-expression network using the precily 2.0 dataset.

B) Protein language model

Evolutionary Scale Modelling (ESM) is a frontier deep learning model that is meant for analyzing protein sequences. For instance, one of the specific models in this category is ESM-1b which uses transformer architecture to handle sequential data like proteins. In addition, ESM-1b processes large datasets allowing it to learn complicated amino acid patterns and interconnections among protein sequences enabling it to make predictions of diverse properties of proteins [16]. ESM1b model which calculates the log likelihood ratio for all missense mutations in a protein, processing the entire protein sequence once to obtain the desired output.

Following below are the steps involved in ESM1b working:

Training Data used in ESM1b:

Protein Sequences: ESM-1b model uses a large dataset of protein sequences from many different kinds of organisms. In this data, there are proteins with various functions.

Model Architecture for ESM1b:

Transformer-Based: The ESM-1b uses a type of neural network architecture called transformer which are good at capturing long-range relationships in sequences which is essential when it comes to understanding protein functions.
Masked Language Modeling (MLM):
Masking: During training, ESM-1b undergoes a process called masked language modeling (MLM). Random sections of the protein sequence are masked by hiding out these amino acids.

Computing the LLR score for missense mutation:

The missense effect scores are calculated by ESM1b from the wild-type amino acid sequence, which provides the log likelihood of each of the 20 standard amino acids (including the wild-type amino acid) at every position in a protein sequence.

The LLR score is relative and it differentiates two conditions; one when there is possibility to have an impact caused by mutations and another case where it may not be.

Higher positive scores suggest a significant impact, while lower or negative scores suggest a neutral effect.

Importance of LLR Score

a) Positive LLR Score: Indicates that this mutation might significantly alter structural or functional properties of proteins and could cause disease.

b) Negative LLR Score: Suggests that this mutation most likely has no adverse effects on protein activity.

c) Near-Zero LLR Score: Implies nearly equal likelihood of original and mutated sequences, therefore benign status for mutations.

2) Experiment

A) **Antibiotic**

Kanamycin is an aminoglycoside bacteriocidal antibiotic agent, used to treat infections caused by gram-negative bacteria and some gram-positive bacteria. It is a mixture of three chemicals: Kanamycin A, B, and C in which A is the main component. Kanamycin binds to the 30S rRNA irreversibly and hence prevents proper protein synthesis. The incorporation of incorrect amino acids makes a non-functional protein. Accumulation of non-functional proteins disrupts the bacteria's vital processes causing death. It doesn't work against anaerobic bacteria, fungi, or viruses.

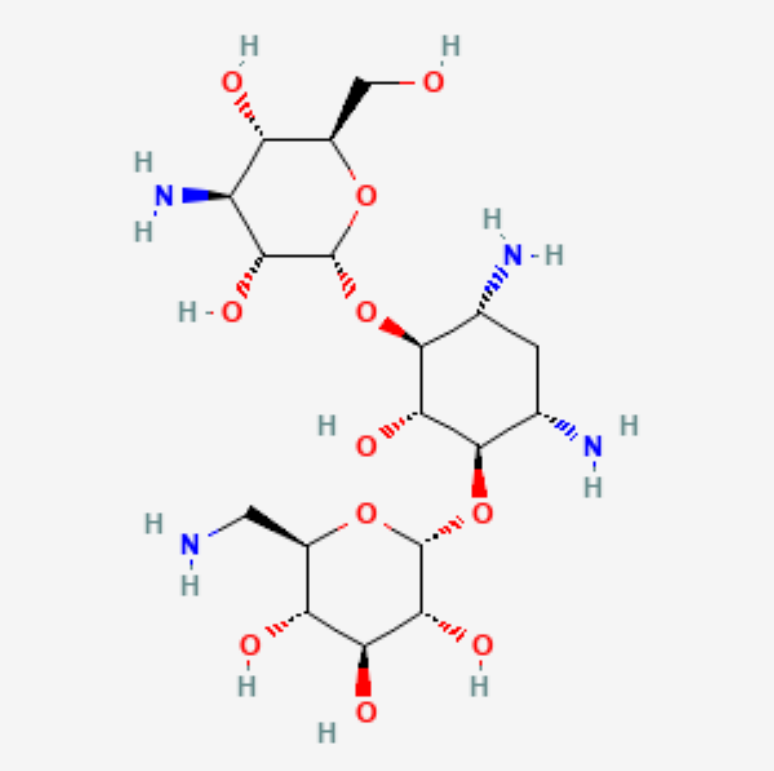


Figure2

https://pubchem.ncbi.nlm.nih.gov/compound/Kanamycin

B) Media

Luria Broth (LB)

Principle :

Luria Broth (LB) is a commonly used nutrient medium for the growth of bacteria, especially *Escherichia coli*. It provides the necessary nutrients and optimal conditions for bacterial growth [17].

Composition :

Reagent	**For 100 mL**	**For 500 mL**	**For 1000 mL**
Tryptone	1 g	5 g	10 g
NaCl	1 g	5 g	10 g
Yeast extract	0.5 g	2.5 g	5 g
Agar	1.5 g	7.5 g	15 g

Table 1

The typical pH of LB is adjusted to around 7.0 before sterilization, which is optimal for most bacteria. The components are dissolved in distilled water, and the solution is sterilized by autoclaving at 121°C for 15-20 minutes to eliminate any contaminants [18].

To prepare 1 liter of standard LB broth: Dissolve 10 g of tryptone, 5 g of yeast extract, and 10 g of sodium chloride in distilled water. Adjust the total volume to 1 liter with distilled water. If necessary, adjust the pH to 7.0. Sterilize the solution by autoclaving. LB medium is essential in microbiology for bacterial growth and maintenance, cloning, and protein expression experiments[19].

M9 minimal media

Principle :

M9 minimal media is a defined medium commonly employed in microbiology to culture bacteria, especially *Escherichia coli* (*E. coli*), in a controlled environment. The term "minimal" refers to its composition containing only the essential nutrients necessary for bacterial growth. Its formulation typically includes:

Composition:

M9 (5x) Salts

Component	Quantity
Na_2HPO_4	32 grams
KH_2PO_4	7.5 grams
NaCl	1.25 grams
NH_4Cl	2.5 grams
Water	500 ml

Table 2

500 ml of M9 minimal media:

Component	Quantity
$MgSO_4$ (1M)	1 ml
$CaCl_2$ (1M)	0.05 ml
Glucose (20%)	10 ml
Agar	400 ml
M9 salts (5X)	100 ml
Trace elements	0.25 ml

Table 3

Note - 250 microliters of trace elements solution containing iron chloride ($FeCl_3$), zinc sulfate ($ZnSO_4$), copper chloride ($CuCl_2$), manganese sulfate ($MnSO_4$), cobalt chloride ($CoCl_2$), and water [20].

C) Assays

i) Growth curves

A timeline showing how bacteria grow over time is termed as growth curve. There are four main phases:

1. Lag Phase: This is equivalent to the starting line for bacteria; they don't really multiply much since they are adjusting to their new surroundings.

2. Log (Exponential) Phase: It's like when bacteria press on the accelerator pedal and start dividing rapidly, causing a population explosion.

3. Stationary Phase: At some point, things start moving at a steady rate. Perhaps resources become limited or waste materials accumulate. Thus growth slows down so that the number of new and dying off bacterial cells balance each other out.

4. Death Phase: Lastly, if conditions continue deteriorating then more organisms die than are born, and eventually population size shrinks.

ii) MIC

MIC (Minimum Inhibitory Concentration) is a key term in microbiology specifically with reference to antibiotic response among bacteria. It refers to the lowest concentration of an antibiotic that effectively inhibits the visible growth of bacteria in a test environment.

Detailed explanations are as follows:

a) Bacteria antibiotic interaction: In the presence of antibiotics bacteria growth gets affected. Different concentrations of antibiotics have variable effects on the growth of bacteria. At lower concentrations depending upon other factors bacteria might be able to grow but at higher concentrations, their growth gets inhibited completely.

b) Determination of MIC: To find the MIC, bacteria are exposed to different concentrations of antibiotics. The minimum concentration of antibiotic in which bacteria can able to grow is noted as the MIC.
c) MIC result interpretation: The MIC value is very important to determine the effect of an antibiotic against a particular strain of bacteria. A lower MIC indicates that the bacteria are more susceptible to the antibiotics whereas a higher MIC suggests that a higher concentration of antibiotic is needed to inhibit the growth of bacteria.

iii) INT/PMS

Principle:

In the assay setup, bacterial cultures grown in media (typically with an optical density, OD, between 0.8 to 1) are mixed with INT and PMS. The bacteria, being metabolically active, have high levels of NADH. This NADH transfers electrons to PMS.

a) PMS (Phenazine Methosulfate): Acting as an electron carrier, PMS facilitates electron transfer from NADH to INT.
b) INT (Iodophenyl Nitrophenyl Tetrazolium Chloride): This compound serves as an electron acceptor and forms a purple formazan product upon reduction.
c) Enzyme Activity: The assay targets enzymes utilizing NADH or NADPH as electron donors, including various dehydrogenases and reductases.

Measurement:

Following the reaction, the formazan precipitate is solubilized, and its absorbance is measured using ImageJ. An increased intensity value in ImageJ states high enzyme activity or a highly metabolically active colony [21].

Composition :

Component	Solution
INT(1mM)	26.5µl
PMS(2.5mM)	0.3µl
Water	73µl
Total	100µl

Table 4

D) Experimental Work Flow for *E. coli* MG1655 Strain Adaptation to Kanamycin

The purpose of this study is to adapt the *E. coli* strain MG1655 to increasing concentrations of kanamycin and identify most potent colonies with the help of growth curve analysis and metabolic activity tests.

A. Growth Curve Assay for D Population

1. Initial Culture and Passage:

Plating: The founder bacteria (*E. coli* MG1655) are spread out on agar plates containing LB + 0.006mg/ml kanamycin (D1). This is a concentration that exerts selective pressure without killing the cells instantly. Plates are incubated at a temperature of 37°C till colonies develop, which can be within a 12-16 hours period.

2. Growth Assay:

Colony Selection: When colonies appear, the colonies are picked and inoculated into a 96-well plate containing LB with similar antibiotic concentration.

Growth Monitoring: Optical density at 600 nm was measured every half an hour over 16 hours using a plate reader so as to construct detailed growth curves.

Data Analysis: Each colony's growth curve (OD vs time) will be plotted, and AUC calculated which quantifies total growth; hence identifying the fittest colony having the highest AUC and earliest entry into the log phase indicating higher growth efficiency and metabolic activity

3. Subsequent Passages:

Increasing Kanamycin Concentration: The process is repeated with the selected colony, increasing the kanamycin concentration stepwise as follows: D2 = 0.008 mg/ml, D3 = 0.01 mg/ml, D4 = 0.04 mg/ml, and D5 = 0.08 mg/ml. For each passage,

the growth assay is conducted, and the fittest colony is selected for the next passage.

B. INT/PMS Assay for R Population

1. Initial Culture and Passage:

Plating: *E. coli* MG1655 cells are spread plated on M9 minimal media agar plates containing 0.006 mg/ml kanamycin (P1). Minimal media is used to impose additional nutritional stress, further selecting for robust colonies. The plates are incubated at 37°C for 12-16 hours until colonies are visible.

2. Spot Assay for Growth:

Preparation:

a) Colonies are grown until they reach an OD of ~0.8 to 1.0.
b) Small portions of each colony are picked and resuspended in 200 µl LB containing the current kanamycin concentration.
c) The cultures are covered with parafilm and incubated for 6-8 hours at 37°C to ensure sufficient growth.

3. INT/PMS Spot Assay:

Reaction Setup:

a) Add 10 µl of the grown culture to 5 µl of the assay solution containing INT (Iodonitrotetrazolium chloride) and PMS (1-Methoxy-5-methylphenazinium methyl sulfate).
b) INT serves as an electron acceptor, which, when reduced by cellular NADH/NADPH, forms a red/violet formazan complex, indicating metabolic activity.

Spotting and Visualization:

a) Spot 10 µl of the reaction mixture onto Whatman filter paper 3 (catalogue number 1003090).

b) The intensity of the red/violet color indicates the metabolic activity of each colony.
c) The colony with the darkest color intensity is selected for the next passage, as it is the most metabolically active.

4. Subsequent Passages:

Increasing Kanamycin Concentration: Similar to the D population, increase the kanamycin concentration stepwise as follows: P2 = 0.01 mg/ml, P3 = 0.05 mg/ml, P4 = 0.06 mg/ml, P5R1 = 0.2 mg/ml, and P5R3 = 0.01 mg/ml.

Repeat the growth and INT/PMS assays to identify the fittest colony at each stage.

C. Minimum Inhibitory Concentration (MIC) Determination

1. Preparation:

Inoculation: Grow the selected potent colony in 5 ml LB containing kanamycin until the OD reaches ~0.8 to 1.0.

2. MIC Assay:

a) Setup: Prepare a range of kanamycin concentrations in 200 µl LB in a 96-well plate.
b) Incubation: Inoculate each well with the potent colony and incubate overnight in a plate reader to continuously measure growth or take endpoint readings.
c) Data Collection: For endpoint readings, measure the OD at the start (T0) and at the endpoint (T-end). Subtract the T0 OD from the T-end OD to determine the growth at each concentration. Plot the data using Prism software to determine the MIC, the lowest concentration of kanamycin that inhibits visible growth.

Replicates:

Perform the MIC assay in triplicates to ensure accuracy and reproducibility.

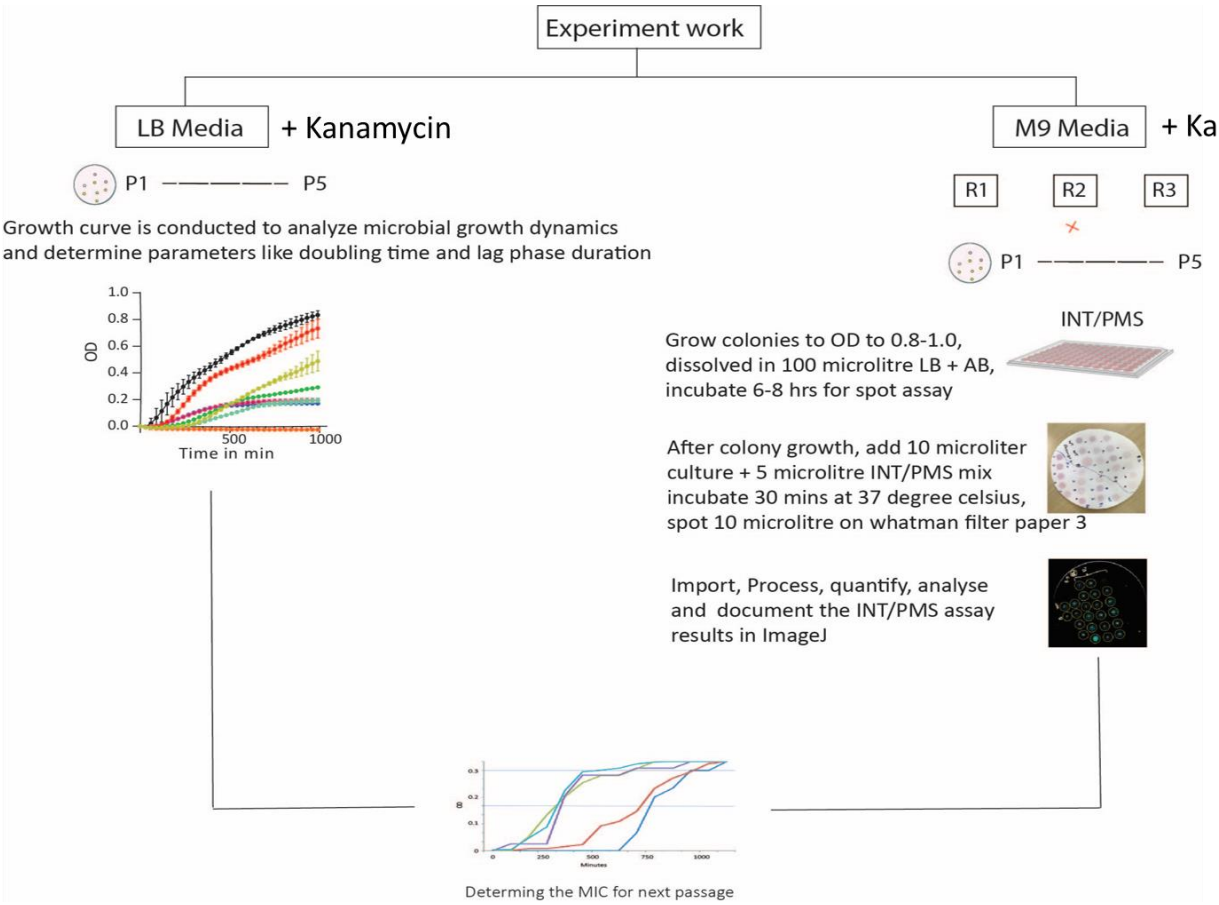

Experiment work
LB Media
+ Kanamycin
M9 Media
+ Kanamycin
P1 ——————— P5
R1
R2
R3
P1 ——————— P5
Growth curve is conducted to analyze microbial growth dynamics and determine parameters like doubling time and lag phase duration
OD
Time in min
INT/PMS
Grow colonies to OD to 0.8-1.0, dissolved in 100 microlitre LB + AB, incubate 6-8 hrs for spot assay
After colony growth, add 10 microliter culture + 5 microlitre INT/PMS mix incubate 30 mins at 37 degree celsius, spot 10 microlitre on whatman filter paper 3
Import, Process, quantify, analyse and document the INT/PMS assay results in ImageJ
Determing the MIC for next passage

Figure 3

Images for the experimental outcomes

Founder

Figure 4

All Passages: Example image

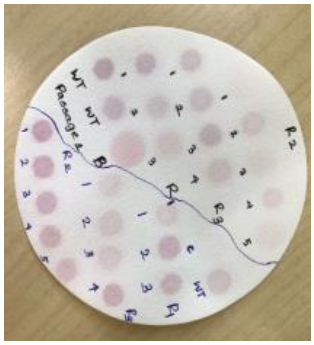


Figure 5

Control plate : WT + PBS
Passage plate
INT/PMS assay

3) Bioinformatics Pipeline for Variant Calling from WGS Sequencing Data

Application: Whole Genome Sequencing (WGS)
Library Type: Bacterial Culture Plate
Platform: Illumina NovaSeq 6000, Paired-End 150 bp Reads

1. Quality Control of Data (QC):

FastQC [22] and MultiQC [23] were used to assess the raw sequencing data's quality. The following metrics were assessed: base call quality distribution, GC content, percentage of bases with quality scores above Q30, and adapter contamination

Sample	Forward reads	Reverse reads
D1	3190502	3190502
D2	4755035	4755035
D3	3697697	3697697
D4	3328196	3328196
D5	4276686	4276686

Table 5

Sample	Forward reads	Reverse reads
P1R3	2898146	2898146
P2R3	4399483	4399483
P3R3	3540935	3540935
P4R3	4005021	4005021
P5R3	5452821	5452821

Table 6

Sample	Forward reads	Reverse reads
P1R1	3316578	3316578
P2R1	3494230	3494230
P3R1	3203512	3203512
P4R1	3591870	3581870
P5R1	4598934	4598934

Table 7

Adapter Sequences:
Read 1: AGATCGGAAGAGCACACGTCTGAACTCCAGTCA
Read 2: AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT

2. Pre-processing Data

To remove adapter sequences and low-quality bases from raw sequence reads we are using fastp version 0.12.4 [24] using its default settings.

3. Calling Variants and Aligning Reads

Alignment of pre-processed reads against their respective reference genomes was carried out by BWA-MEM algorithm [25]. Post-alignment processing involved removal of PCR duplicates using Picard [26] and Sambamba [27]. To enhance variant calling accuracy we performed base quality score recalibration.

For variant calling FreeBayes variant caller was utilized with minimum quality score threshold set at 20 for filtering purposes.

4. Annotating Variants

The annotated variants were then processed with SnpSift [30] so as to convert them into tab delimited text files that could facilitate other analysis or interpretation purposes in VCF format files.

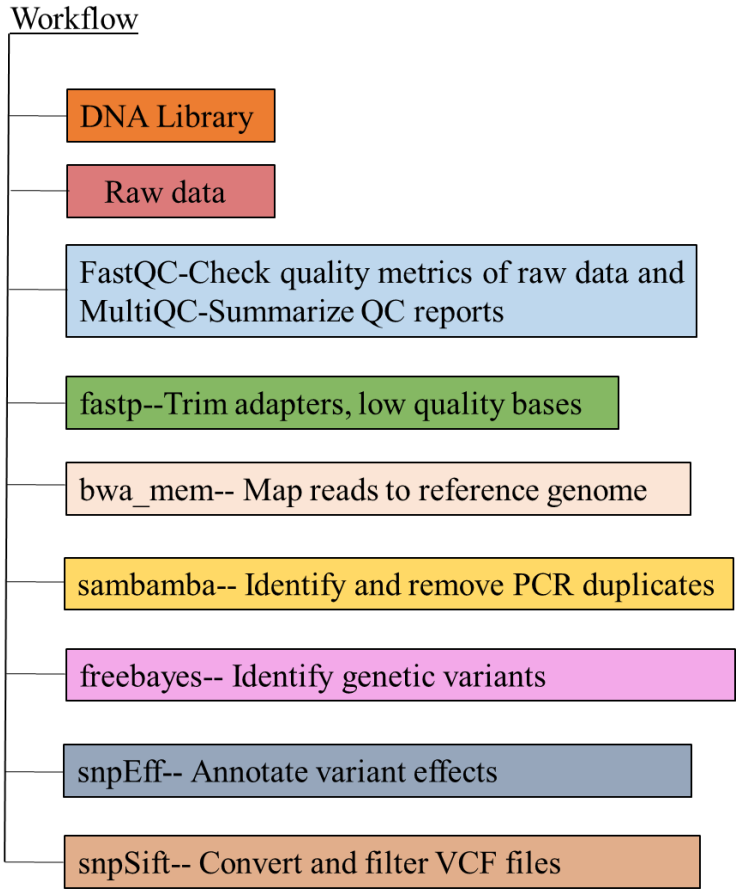


Figure 6

5) Scripts

To count raw reads

```bash
#!/bin/bash

# Define the directory containing the FASTQ.gz files
directory="/home/bernadettem/nancy_thesis/pilot_analysis/Raw_Reads"

# Define the output file to save counts
output_file="/home/bernadettem/nancy_thesis/pilot_analysis/raw_reads_count.txt"

# Clear the contents of the output file or create it if it doesn't exist
> "$output_file"

# Loop through all .fastq.gz files in the directory
for file in "$directory"/*.fastq.gz; do
    # Extract the file name without the directory path and extension
    filename=$(basename "$file" .fastq.gz)

    # Count the number of reads in the file
    count=$(zcat "$file" | grep -c "^@.*$")

    # Print the file name and read count to the output file
    echo "File: $filename, Read Count: $count" >> "$output_file"
done
```

Script 1

To run FastQC and MultiQC

```bash
#!/bin/bash

# # # Run FastQC on all raw reads
fastqc -t 24 -o /home/bernadettem/nancy_thesis/pilot_analysis/fastqc_raw_17_04_2024 /home/bernadettem/nancy_thesis/pilot_analysis/Raw_Reads/*.fastq.gz

# Run Multiqc on all raw reads
multiqc /home/bernadettem/nancy_thesis/pilot_analysis/fastqc_raw_17_04_2024
multiqc -o /home/bernadettem/nancy_thesis/pilot_analysis/multiqc_data_raw_reads_17_04_2024 /home/bernadettem/nancy_thesis/pilot_analysis/fastqc_raw_17_04_2024

##########################################################################################################################

# run Fastqc on fastp data
fastqc -t 24 -o /home/bernadettem/nancy_thesis/pilot_analysis/fastp_fastqc_17_04_2024 /home/bernadettem/nancy_thesis/pilot_analysis/fastp_17_04_2024/*.fastq.gz

# Run Multiqc on all raw reads
multiqc /home/bernadettem/nancy_thesis/pilot_analysis/fastp_fastqc_17_04_2024
multiqc -o /home/bernadettem/nancy_thesis/pilot_analysis/multiqc_data_fastP_reads_17_04_2024 /home/bernadettem/nancy_thesis/pilot_analysis/fastp_fastqc_17_04_2024
```

Script 2

To run fastp

```
# Specify the path to the adapter fasta file
adapter_fasta="/home/bernadettem/nancy_thesis/pilot_analysis/adapters.fa"

# Specify the number of threads
threads=42

# Specify the input and output directories
input_dir="/home/bernadettem/nancy_thesis/pilot_analysis/Raw_Reads"
output_dir="/home/bernadettem/nancy_thesis/pilot_analysis/fastp_17_04_2024"


# Process each pair of input files
for file_prefix in D1-1 D2-1 D3 D4 D5 F P1R1 P1R3 P2R1 P2R3 P3R1 P3R3 P4R1 P4R3 P5R1 P5R3 WT7; do
    # Define input and output file paths
    input_r1="${input_dir}/${file_prefix}_R1.fastq.gz"
    input_r2="${input_dir}/${file_prefix}_R2.fastq.gz"
    output_r1="${output_dir}/${file_prefix}_fastp_paired_R1.fastq.gz"
    output_r2="${output_dir}/${file_prefix}_fastp_paired_R2.fastq.gz"
    report_html="${output_dir}/${file_prefix}_fastp_report.html"
    report_json="${output_dir}/${file_prefix}_fastp_report.json"

    # Run fastp
    fastp --adapter_fasta "$adapter_fasta" \
          --low_complexity_filter \
          --dedup \
          --trim_poly_g \
          --trim_poly_x \
          --detect_adapter_for_pe \
          --overrepresentation_analysis \
          --correction \
          --cut_right \
          --thread "$threads" \
          --html "$report_html" \
          --json "$report_json" \
          -i "$input_r1" \
          -I "$input_r2" \
          -o "$output_r1" \
          -O "$output_r2"

    echo "Processed: $file_prefix"
done
```

Script 3

To run bwa mem for indexing

```
#!/bin/bash

# Set the input reference file and index prefix
input_reference="/home/bernadettem/nancy_thesis/pilot_analysis/bwa_indexing_17_04_2024/GCF_000005845.2_ASM584v2_genomic.fna"
index_prefix="/home/bernadettem/nancy_thesis/pilot_analysis/bwa_indexing_17_04_2024/index_prefix"

# Run the bwa index command
bwa index -a bwtsw "$input_reference" "$index_prefix"
```

Script 4

To run bwa mem for mapping with the reference genome

```
#!/bin/bash

# Set reference genome
reference="/home/bernadettem/nancy_thesis/pilot_analysis/bwa_indexing_17_04_2024/GCF_000005845.2_ASM584v2_genomic.fna"

# Output directory
output="/home/bernadettem/nancy_thesis/pilot_analysis/bwa_alignment_17_04_2024"

# Loop over all FASTQ files in the directory
for file_1 in /home/bernadettem/nancy_thesis/pilot_analysis/fastp_17_04_2024/*_fastp_paired_R1.fastq.gz; do
    # Get corresponding file_2
    file_2="${file_1/_R1/_R2}"

    # Extract sample name
    sample=$(basename "$file_1" _fastp_paired_R1.fastq.gz)

    # Run bwa mem command
    bwa mem -t 8 \
            -k 19 \
            -w 100 \
            -d 100 \
            -r 1.5 \
            -c 10000 \
            -A 1 \
            -B 4 \
            -O 6 \
            -E 1 \
            -L 5 \
            -U 9 \
            -v 3 \
            -M \
            "$reference" "$file_1" "$file_2" > "${output}/${sample}_alignments.sam"
done
```

Script 5

To run samtools

```bash
#!/bin/bash

# Output directory
output="/home/bernadettem/nancy_thesis/pilot_analysis/bwa_alignment_17_04_2024"

# Loop over all SAM files in the output directory
for sam_file in "${output}"/*_alignments.sam; do
    # Convert SAM to BAM
    samtools view -@ 8 -bS "$sam_file" > "${sam_file%.sam}.bam"

    # Filter the data by removing the unmapped reads
    samtools view -b -F 0xc "${sam_file%.sam}.bam" -o "${sam_file%.sam}_mapped.bam"

    # Sort the mapped data
    samtools sort -@ 8 "${sam_file%.sam}_mapped.bam" -o "${sam_file%.sam}_sorted.n.bam"

    # Sort the mapped data on the basis of query name
    samtools sort -n "${sam_file%.sam}_sorted.n.bam" -o "${sam_file%.sam}_sorted.queryname.bam"

    # Do the fixmate
    samtools fixmate -m "${sam_file%.sam}_sorted.queryname.bam" "${sam_file%.sam}_fixmate.bam"

    # Sort the fixmate data
    samtools sort -n -@ 8 "${sam_file%.sam}_fixmate.bam" -o "${sam_file%.sam}_fixmate.sorted.bam"

    # Sort the fixmate data according to the coordinate
    samtools sort "${sam_file%.sam}_fixmate.sorted.bam" -o "${sam_file%.sam}_fixmate.sorted.coordinate.bam"

    # Mark and remove the duplicates
    samtools markdup -r -@ 8 "${sam_file%.sam}_fixmate.sorted.coordinate.bam" "${sam_file%.sam}_dedup.bam"

    # Index sorted BAM file
    samtools index "${sam_file%.sam}_dedup.bam"

    # Generate alignment summary
    samtools flagstat "${sam_file%.sam}_dedup.bam" > "${sam_file%.sam}_dedup.txt"
done
```

Script 6

To run freebayes for VCF calling

```
#!/bin/bash

# for bam_file in /home/bernadettem/nancy_thesis/pilot_analysis/bwa_alignment_17_04_2024/*_alignments_dedup.bam; do
#     output_vcf="/home/bernadettem/nancy_thesis/pilot_analysis/vcf_filtered/$(basename "${bam_file%.*}")_filtered.vcf"
#     freebayes --min-base-quality 20 -f /home/bernadettem/nancy_thesis/pilot_analysis/bwa_indexing_17_04_2024/
#      GCF_000005845.2_ASM584v2_genomic.fna -b "$bam_file" --vcf "$output_vcf"
# done

# for bam_file in /home/bernadettem/nancy_thesis/pilot_analysis/bwa_alignment_17_04_2024/*_alignments_dedup.bam; do
#     output_vcf="/home/bernadettem/nancy_thesis/pilot_analysis/vcf_unfiltered/$(basename "${bam_file%.*}")_unfiltered.vcf"
#     freebayes -f /home/bernadettem/nancy_thesis/pilot_analysis/bwa_indexing_17_04_2024/
#      GCF_000005845.2_ASM584v2_genomic.fna -b "$bam_file" --vcf "$output_vcf"
# done


for bam_file in /home/bernadettem/nancy_thesis/pilot_analysis/bwa_alignment_17_04_2024/*_alignments_dedup.bam; do
    output_vcf="/home/bernadettem/nancy_thesis/pilot_analysis/vcf_filtered_mod/$(basename "${bam_file%.*}")_filtered_mod.vcf"
    freebayes --min-base-quality 20 --min-mapping-quality 20 --min-alternate-count 2 --min-coverage 10 --min-alternate-fraction 0.01 -f
    /home/bernadettem/nancy_thesis/pilot_analysis/bwa_indexing_17_04_2024/GCF_000005845.2_ASM584v2_genomic.fna -b
    "$bam_file" --vcf "$output_vcf"
done
```

Script 7

Run snpeff for variant annotation

```
#!/bin/bash

# # Set the path to the SNPeff JAR file
# SNPEFF_JAR="/home/bernadettem/nancy_thesis/snpEff/snpEff.jar"

# # Iterate over all .vcf files in the directory
# for vcf_file in /home/bernadettem/nancy_thesis/pilot_analysis/vcf_filtered/*.vcf; do
#     # Extract the filename without extension
#     filename=$(basename -- "$vcf_file")
#     filename_no_ext="${filename%.*}"

#     # Run SNPeff for each VCF file
#     java -Xmx8g -jar "$SNPEFF_JAR" GCA_000005845 "$vcf_file" > "/home/bernadettem/nancy_thesis
#      /pilot_analysis/vcf_filtered_annotated/${filename_no_ext}.ann.vcf"
# done


# Set the path to the SNPeff JAR file
SNPEFF_JAR="/home/bernadettem/nancy_thesis/snpEff/snpEff.jar"

# Iterate over all .vcf files in the directory
for vcf_file in /home/bernadettem/nancy_thesis/pilot_analysis/vcf_filtered_mod/*.vcf; do
    # Extract the filename without extension
    filename=$(basename -- "$vcf_file")
    filename_no_ext="${filename%.*}"

    # Run SNPeff for each VCF file
    java -Xmx8g -jar "$SNPEFF_JAR" GCA_000005845 "$vcf_file" > "/home/bernadettem/nancy_thesis/pilot_analysis/
    vcf_filtered_mod_annotated/${filename_no_ext}.ann.vcf"
done
```

Script 8

Run snpsift to filter vcf files

```bash
#!/bin/bash


# Define input directory
input_dir="/home/bernadettem/nancy_thesis/pilot_analysis/vcf_filtered_mod_annotated"

# Loop through each .ann.vcf file in the directory
for file in ${input_dir}/*.ann.vcf; do
    # Extract file name without extension
    filename=$(basename -- "$file")
    filename_no_ext="${filename%.*}"

    # Execute SnpSift and awk commands
    java -jar SnpSift.jar extractFields -s ' ' "$file" CHROM POS ID REF ALT FILTER AF AC DP MQ
    ANN[0].ALLELE ANN[0].EFFECT ANN[0].IMPACT ANN[0].GENE ANN[0].GENEID ANN[0].FEATURE ANN[0].FEATUREID >
    "/home/bernadettem/nancy_thesis/pilot_analysis/snpsift_filtered_mod/${filename_no_ext}_output.txt" && awk 'BEGIN {OFS="\t"} {print}'
    "/home/bernadettem/nancy_thesis/pilot_analysis/snpsift_filtered_mod/${filename_no_ext}_output.txt" >
    "/home/bernadettem/nancy_thesis/pilot_analysis/snpsift_filtered_mod/${filename_no_ext}_output.csv"
done
```

Script 9

Chapter 4

The technique implemented

After the completion of the variant calling of sequencing data, we will proceed with a series of advanced computational and experimental steps to decipher gene interactions, pathway involvement and the effect of mutations. The following is the detailed procedure:

1. RWR implementation: We will employ Random Walk with Restart to identify the top 100 genes that strongly interact with the multi-seed gene having a high score. This technique helps to identify the most influential nodes within the gene interaction network, ensuring the capture of key genes that play significant roles in biological processes.

2. Pathway Analysis

Preparing the Gene List: After obtaining the top 100 interacting genes from the gene interaction network analysis, compile these genes into a single list formatted for input into ShinyGO.

a) Accessing ShinyGO: Navigate to the ShinyGO web application at ShinyGO.

b) Inputting Gene List: On the ShinyGO main interface, enter the list of the top 100 interacting genes. Ensure that the gene identifiers (such as gene symbols or Ensembl IDs) are in a format supported by ShinyGO.

c) Selecting Organism and Background: Choose the relevant organism for your study (e.g.,*Escherichia coli* MG1655). Optionally, select a background gene

set if you want to compare your gene list against a specific background other than the default whole genome.

d) Performing Enrichment Analysis: Click on the "Submit" button to start the enrichment analysis. ShinyGO will analyze the input gene list and provide a summary of enriched pathways. The results will include various types of pathway enrichment analyses, including Gene Ontology (GO) terms, KEGG pathways, and Reactome pathways.

Interpreting Results:

Pathway Visualization:
ShinyGO provides visual representations of enriched pathways, including pathway diagrams and enrichment plots.
Significance Scores:
Examine the p-values, adjusted p-values (e.g., FDR), and enrichment scores to determine the significance of each pathway.
Functional Insights:
Review the biological processes, molecular functions, and cellular components associated with your gene list to gain functional insights.
Exporting Data:
Download the results, including lists of enriched pathways and associated genes, in various formats (e.g., CSV, PDF) for further analysis or presentation.

3. Downstream gene analysis

Log-Likelihood Ratio (LLR) Score Calculation: To evaluate the impact of specific mutations on protein structure and function, we will calculate the Log-Likelihood Ratio (LLR) score. This score helps differentiate between mutations that are likely to have detrimental effects on the protein's stability or function and those that are neutral. By integrating LLR scores, we can prioritize mutations for further experimental validation.

4. Supporting literature - Literature Survey and Pathway Significance

Comprehensive Literature Review: A thorough literature survey will be conducted to assess the importance of the identified pathways, particularly in relation to antibiotic resistance mechanisms. We will focus on studies that link these pathways to resistance against kanamycin, an aminoglycoside antibiotic. This review will help contextualize our findings within the broader scientific knowledge, highlighting potential mechanisms through which these pathways contribute to resistance.

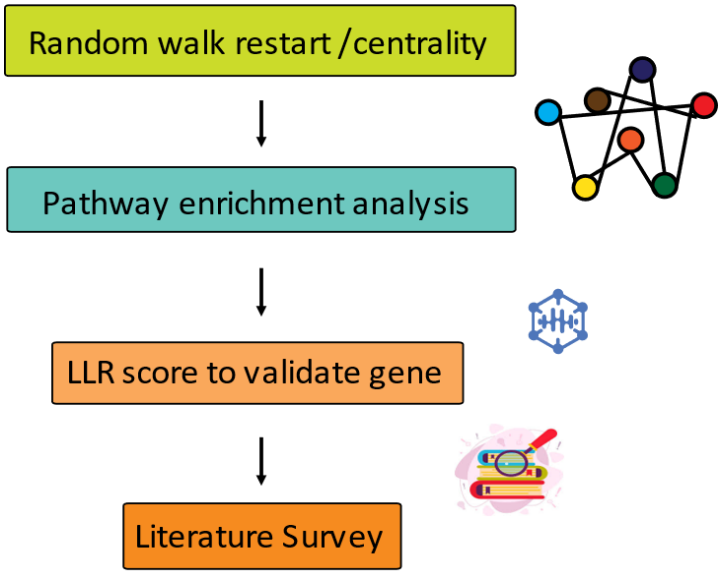


Figure 7

Integrating Random Walk with Restart for Analyzing *E. coli* Gene Networks

In biological research, understanding the complex interactions within gene networks is crucial. One effective method to explore these interactions is the Random Walk with Restart (RWR) algorithm. RWR simulates a particle that moves randomly through the network but intermittently restarts from specific seed genes. This prevents the particle from getting stuck and allows it to explore the network more comprehensively.

In our study, we applied RWR to an integrated network composed of three layers:

Protein interactions (STRING network)
Gene interactions specific to *E. coli* (EcoliNet)
Co-expression patterns (WGCNA network)
These layers provide varied insights into gene behaviors, and their integration offers a holistic view of gene interactions across different contexts.

Seed Genes for RWR Analysis
We designated specific seed genes for the D and R passages:

D passage seed genes:
ybjI, thyA, yrdA, yihD, cpxA, rpsL, metQ, and *trmA.*
R passage seed genes:
cyoA, yajR, crr, hemL, hemA, tabA, gluQ, cyoB, and *atpG.*

By applying RWR to this network, we identified genes closely connected to our seed genes. This analysis highlights the influence exerted by the seed genes throughout the network, aiding in the understanding of gene co-regulation and functional relationships.

Code to find the seed gene

```r
#install.packages("VennDiagram")
#library(VennDiagram)
# Import dataset
library(readxl)
filtered_data_all <- read_excel("/home/bernadettem/nancy_thesis/multiplexing_homogenous/multiplexing_indels/filtered_data_all.xlsx",
                                sheet = "ms_including_indels")
View(filtered_data_all)
df <- filtered_data_all # make a copy

################################################################################
 # Filter
 D <- df[grepl("^D", df$Label), ] # D alone
 R1 <- df[grepl("R1", df$Label), ] # R1 alone
 R3 <- df[grepl("R3", df$Label), ] # R3 alone
 # Count the number of unique genes in each subset
 D_unique <- unique(D$`ANN[0].GENE`) # Assuming the gene column is called "Gene"
 R1_unique <- unique(R1$`ANN[0].GENE`)
 R3_unique <- unique(R3$`ANN[0].GENE`)

# # Find unique genes in each set
 D_alone <- setdiff(D_unique, union(R1_unique, R3_unique))
 R1_alone <- setdiff(R1_unique, union(D_unique, R3_unique))
 R3_alone <- setdiff(R3_unique, union(D_unique, R1_unique))
#
#
# # Convert D to a character vector
 D_gene_only <- unlist(D_alone)
 R1_gene_only <- unlist(R1_alone)
 R3_gene_only <- unlist(R3_alone)
#
# # Convert vectors to character vectors and unlist them
 D_gene_only <- unlist(as.character(D_alone))
 R1_gene_only <- unlist(as.character(R1_alone))
 R3_gene_only <- unlist(as.character(R3_alone))
#
# # Define the file paths for saving
 D_file <- "D_gene_only.txt"
 R1_file <- "R1_gene_only.txt"
 R3_file <- "R3_gene_only.txt"
#
# # Save vectors as .txt files
 writeLines(D_gene_only, D_file)
 writeLines(R1_gene_only, R1_file)
 writeLines(R3_gene_only, R3_file)
#
# # Optionally print confirmation messages
 cat("D_gene_only saved to:", D_file, "\n")
 cat("R1_gene_only saved to:", R1_file, "\n")
 cat("R3_gene_only saved to:", R3_file, "\n")
```

Script 10

Further to run rwr we use the code from
https://github.com/alberto-valdeolivas/RWR-MH.git

Code to plot the rwr result

```r
library(igraph)

# Read gene data from file
gene_data <- read.table("/home/bernadettem/nancy_thesis/multiplexing_homogenous/multiplexing_indels/top_30_geneR3_passage.gr",
                        header = TRUE, stringsAsFactors = FALSE)

# Create an igraph graph object
graph <- graph_from_data_frame(gene_data, directed = TRUE)

# Define the colors for nodes
node_colors <- rep("yellow", vcount(graph))  # Set default color to yellow for all nodes
highlighted_genes <- c("gluQ","cyoB","atpG")   # D passage
highlighted_index <- which(V(graph)$name %in% highlighted_genes)
node_colors[highlighted_index] <- "red"  # Set color to red for highlighted genes

# Set plot size
options(repr.plot.width=10, repr.plot.height=10)  # Adjust plot size as needed

# Plot the graph
plot_path <- "/home/bernadettem/nancy_thesis/multiplexing_homogenous/multiplexing_indels/top_30_R3.pdf"
#plot_path <- "/home/bernadettem/nancy_thesis/multiplexing_homogenous/multiplexing_indels/R1_100_gene.pdf"
#plot_path <- "/home/bernadettem/nancy_thesis/multiplexing_homogenous/multiplexing_indels/R3_100_gene.pdf"
pdf(plot_path)  # Uncomment this line to save the plot as PDF
plot(
  graph,
  layout = layout_nicely(graph),
  vertex.color = node_colors,
  vertex.label.color = "black",  # Set the color of vertex labels to black
  edge.arrow.size = 0  # Set edge arrow size to zero for undirected edges
)
dev.off()
```

Script 11

Pathway Analysis Using ShinyGO

ShinyGO is a valuable tool for understanding the functional roles of genes through Gene Ontology (GO) annotations. By using RWR, we identified the top 100 genes that exhibit robust interactions with the designated seed genes. Centrality scores provide quantitative insights into these genes' importance within biological networks. These identified genes were then analyzed using ShinyGO to explore their involvement in various biological pathways, particularly those exclusive to the D passage under antibiotic stress conditions.

Procedure:

Identify top 100 genes using RWR and centrality scores.
Analyze these genes in ShinyGO to identify pathways specific to the D passage.
Check if the seed genes in the D passage are present in the identified pathways.
If any seed genes are present, rerun RWR for these seed genes separately.
Merge the top 50 interacting genes from each seed gene and run ShinyGO again to determine their pathway associations.

Evolutionary Significance Measure (ESM) Score

The ESM score assesses the potential functional impact of genetic mutations within proteins, based on evolutionary conservation. A negative ESM score suggests less disruption to protein function, while a positive score indicates a higher likelihood of deleterious effects.

Code to processed the data before performing ESM1b

```r
# Load required libraries
library(openxlsx)

# Load both Excel files
feature_all <- read.xlsx("~/nancy_thesis/esm_result/feature_all.xlsx", sheet = "Sheet2")
DNA_Prt_variant <- read.xlsx("/home/bernadettem/nancy_thesis/esm_result/esm_indel/DNA_Prt_variant.xlsx", sheet = "Sheet1")

# Merge the two data frames based on the 'Gene' column
merged_data <- merge(DNA_Prt_variant, feature_all[, c("Gene", "start", "end", "strand")], by = "Gene", all.x = TRUE)

# Write the merged data to a new Excel file
write.xlsx(merged_data, "~/nancy_thesis/esm_result/esm_indel/DNA_Prt_Strand_variant.xlsx")
```

Script 12

To run esm1b model we use
https://github.com/ntranoslab/esm-variants.git

Analysis of Carry-Forward and Non-Carry-Forward Genes

We categorized genes into carry-forward (appearing consistently in every passage under antibiotic pressure) and non-carry-forward genes (appearing transiently). This comparison between D and R passages provides insights into critical resistance mechanisms.

Steps:

Identify carry-forward and non-carry-forward genes from sequencing data.

```r
# Install and load necessary packages
#install.packages("readxl")
#install.packages("dplyr")
#install.packages("openxlsx")
library(readxl)
library(dplyr)
library(openxlsx)

# Read the Excel file into R
excel_data <- read_excel("all_filtered.xlsx", sheet = "D")

# Group the data by the Label column and summarize the counts of each Gene within each Label
result <- excel_data %>%
  group_by(Label, Gene) %>%
  summarise(counts = n()) %>%
  ungroup()

# Calculate the total count of each gene within each label
result <- result %>%
  group_by(Gene) %>%
  summarise(total_count = sum(counts),
            labels = paste(Label, collapse = ",")) %>%
  ungroup()

# Add the summarized counts and Gene information back to the original data
final_data <- left_join(excel_data, result, by = "Gene")

# Remove duplicates based on the Gene column before saving
final_data <- final_data %>%
  distinct(Gene, .keep_all = TRUE)

# Write the final data to a new Excel file
write.xlsx(final_data, "carryforward_D.xlsx")
```

Script 13

Use the ESM1b model to calculate the LLR (Log-Likelihood Ratio) score for both D and R populations.

Compare the LLR scores to understand differences in antibiotic resistance mechanisms between the D and R passages.

This systematic approach integrates RWR and centrality measures to prioritize genes for further experimental validation, provides pathway insights through ShinyGO, and evaluates the evolutionary impact of genetic mutations, thus enhancing our understanding of gene interactions and resistance mechanisms in *E. coli*.

Chapter 5

Results

1) Experiment

In our study, we conducted a wet lab experiment to examine the growth dynamics of wild type and evolved bacterial populations under varying conditions of antibiotic stress across multiple passages. Specifically, we monitored the growth of wild type colonies and colonies from passages 1 to 5, including the populations designated as D, R1, and R3. All colonies were cultured in Luria-Bertani (LB) broth with the appropriate antibiotic concentration, and their growth was tracked over a 16-hour period with optical density (OD) measurements taken every half hour. The resulting data were normalized and plotted using Prism GraphPad software.

The wild type, which served as the founder cell line and was not subjected to antibiotic stress, showed a characteristic pattern of growth stagnation from passage 1 to passage 5. In contrast, the control group exhibited no growth under the same conditions. For the D population, we assessed growth in both LB without antibiotics (D_LB) and LB with antibiotics (D_AB). We observed that from passage 1 to passage 5, the D population continued to grow robustly, indicating that these bacteria acquired some adaptive mutations that conferred a survival advantage under antibiotic stress.

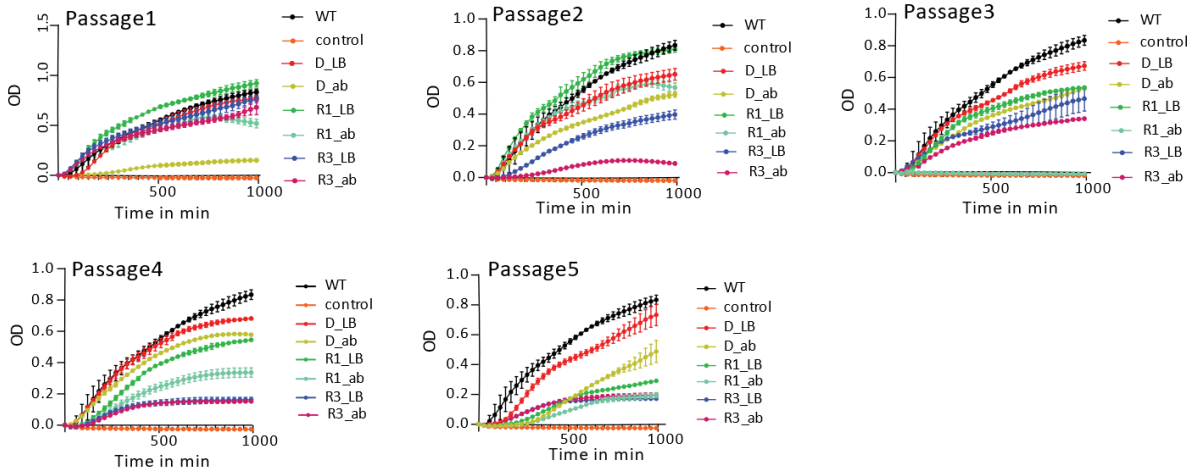
In examining the R populations (R1 and R3), both in the presence and absence of antibiotics, we noticed an initial increase in growth from passage 1 to passage 2. However, as the passages progressed from 3 to 5, a decline in growth was evident. This suggests that during the adaptation process to increasing antibiotic concentrations, certain genetic changes in the R populations resulted in a fitness cost, impeding their growth capacity.

For better visualization, the second plot presents the growth curves at the final time point. This plot confirms the same growth patterns observed: the D population exhibits sustained growth, highlighting their adaptation to antibiotic stress. Conversely, the R populations demonstrate decreased growth from passage 3 to passage 5, indicating a detrimental effect on their fitness due to genetic adaptations.

Our findings imply that the nutrient-rich environment of LB broth used for the D population facilitated the bacteria's ability to withstand antibiotic stress, likely due to the availability of abundant resources necessary for stress response mechanisms. In contrast, the R populations were cultured in M9 minimal media, a defined medium with limited nutrients. This nutrient limitation likely hindered the bacteria's ability to effectively manage and adapt to the antibiotic stress, ultimately crippling their growth over successive passages.

These results highlight the complex interplay between environmental conditions and genetic adaptation in bacterial populations under antibiotic pressure. The differential growth patterns observed in LB and M9 media underscore the critical role of nutrient availability in supporting bacterial survival and adaptation in stressful environments.

Comparative Growth Curve Analysis



Growth curve at final time point

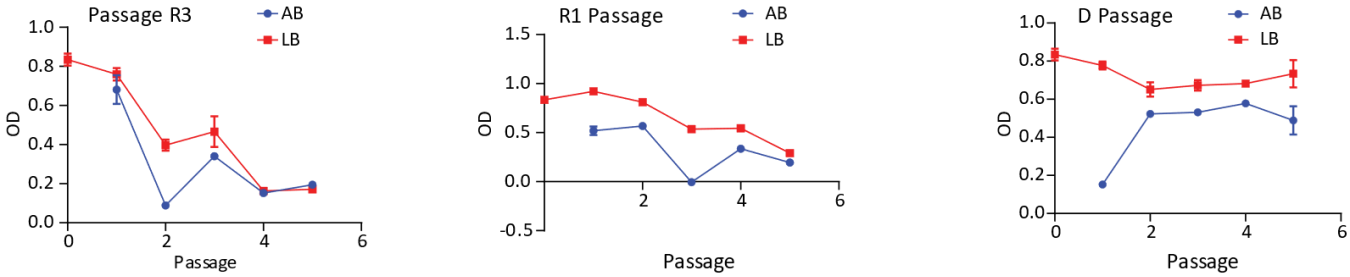


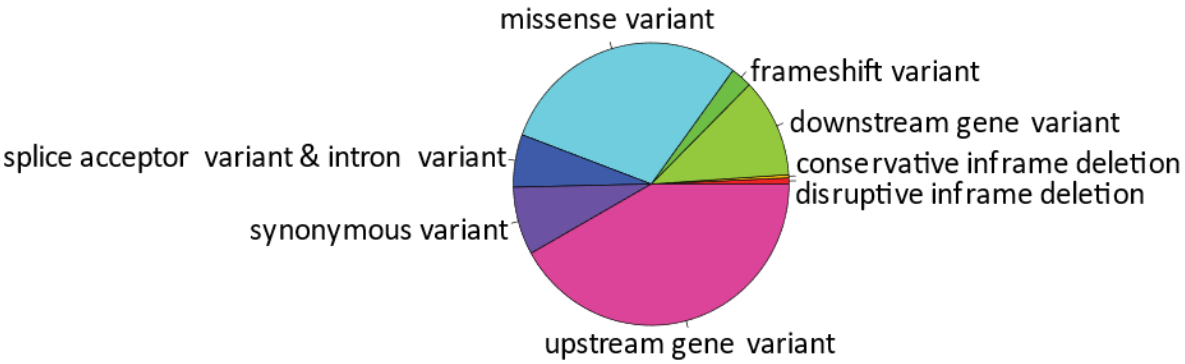
Figure 8

2) Variant Calling

In our bioinformatic analysis pipeline, we processed raw sequencing data to identify variants through variant calling, resulting in the generation of Variant Call Format (VCF) files. These files provided us with mutation counts for each population, namely D, R1, and R3, which we visualized using a bar plot to represent the distribution of mutations across the populations.

Upon annotating the variants, we categorized them into different types, such as single nucleotide polymorphisms (SNPs), insertions, deletions, and complex variants. To illustrate the distribution of these variant types, we utilized a pie chart, offering a comprehensive overview of the mutation landscape within the bacterial populations.

This bioinformatic analysis allowed us to gain insights into the genetic changes occurring within the bacterial populations under antibiotic stress. By characterizing the types and frequencies of mutations, we could elucidate the molecular mechanisms driving adaptation and evolution in response to selective pressure.


missense variant
frameshift variant
downstream gene variant
conservative inframe deletion
disruptive inframe deletion
splice acceptor variant & intron variant
synonymous variant
upstream gene variant

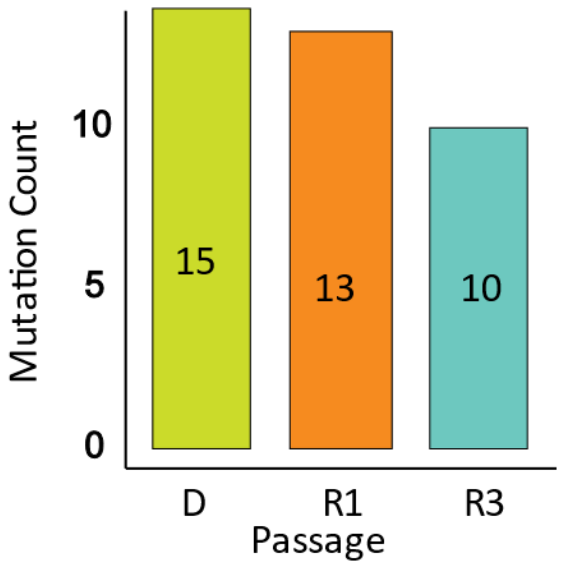

Mutation Count
10
5
0
15
13
10
D
R1
R3
Passage

Figure 9

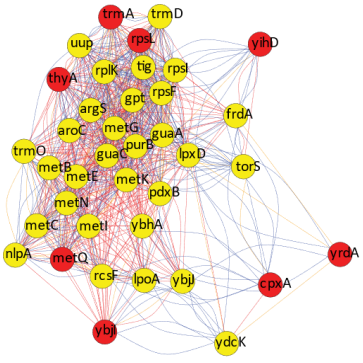
3) RWR and Pathway analysis output

The results of the RWR approach are below to explore the genetic interactions associated with mutated genes from the D, R1, and R3 bacterial populations. Initially, we utilized the mutated genes from these populations as seed genes and performed a Random Walk with Restart (RWR) algorithm on a multiplex network. This algorithm allowed us to identify the top 100 genes that exhibited strong interactions with the seed genes.

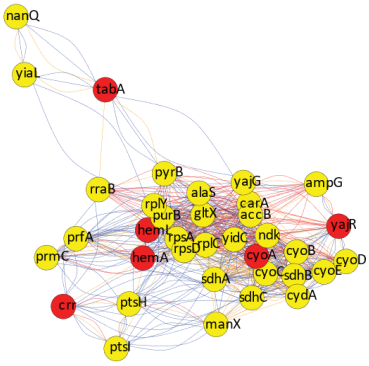
Subsequently, to construct a network plot for visualization, we focused on the top 30 genes

This network-based analysis offers a systems-level perspective on the genetic landscape of bacterial populations under selective pressure. By elucidating the network of genetic interactions surrounding mutated genes, we can uncover key players and regulatory relationships that drive adaptive responses and evolutionary trajectories in response to antibiotic stress.

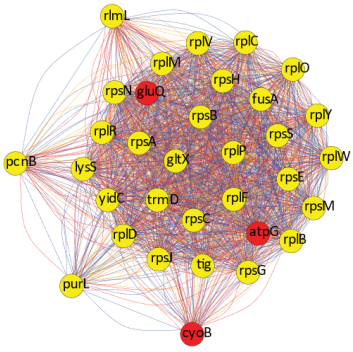
Visualizing the Top 30 Genes with Strong Interactions in the RWR Network



Top 30 genes of D population



Top 30 genes of R1 population



Top 30 genes of R3 population

Figure 10

Pathway enrichment analysis on the top 100 genes identified through Random Walk with Restart (RWR) was performed using a tool called Shiny Gene Ontology (GO), which allowed us to identify biological pathways enriched among the selected genes.

Subsequently, we focused on pathways that were found to be commonly enriched in the D population. By identifying these shared pathways, we aimed to uncover specific biological processes that are potentially implicated in the adaptation and survival of the D bacterial population under antibiotic stress.

Pathway Enrichment Analysis of Top Genes Identified by RWR Using Shiny GO

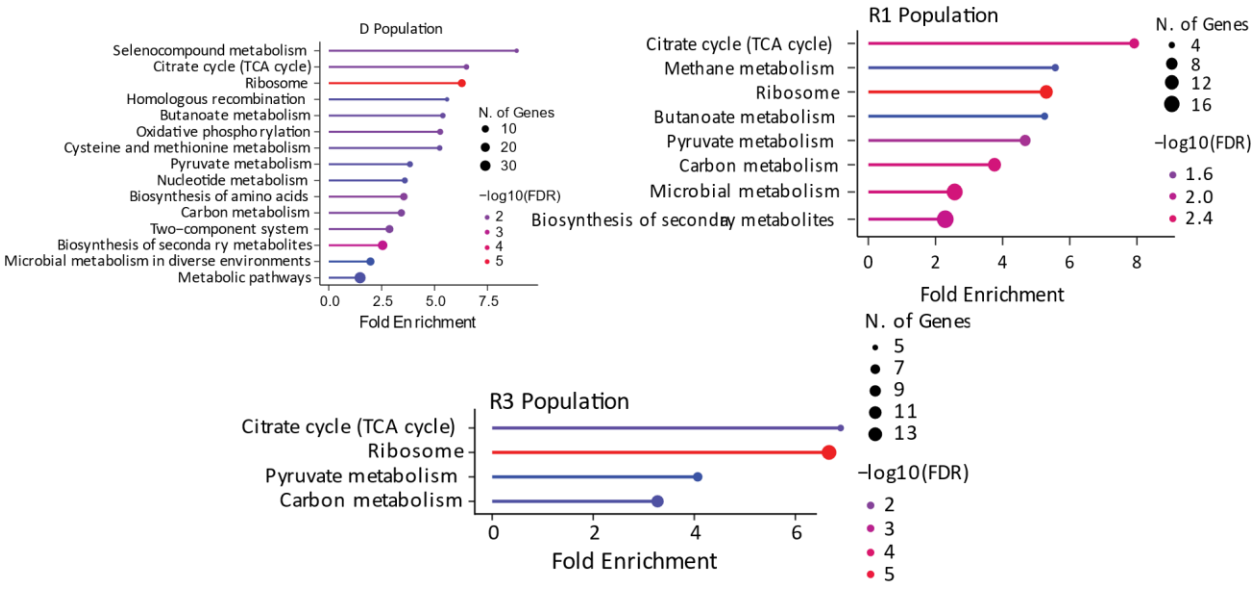


Figure 11

After identifying pathways enriched in the D population through pathway enrichment analysis using Shiny Gene Ontology (GO), we extracted gene information directly from Shiny GO for the selected pathways. From this gene list, we identified three key seed genes: *cpxA, thyA,* and *ybjI*.

To further investigate the genetic interactions associated with these seed genes, we conducted separate Random Walk with Restart (RWR) analyses for each seed gene. This allowed us to identify the top 50 interacting genes associated with each seed gene,but use the top 20 genes which we then visualized in network plots.

By focusing on these specific seed genes and their interacting partners, we aimed to elucidate the genetic networks and potential pathways involved in the adaptive response of the D bacterial population to antibiotic stress. This approach enables us to gain deeper insights into the molecular mechanisms underlying bacterial adaptation and resistance, laying the groundwork for targeted investigations and therapeutic interventions.

Identifying Key Pathways and Genes Associated with Antibiotic Resistance

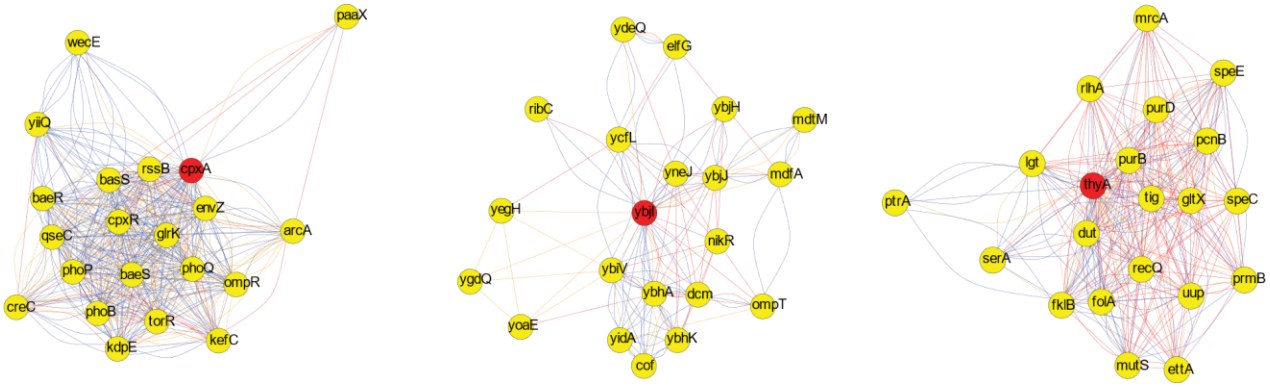


Figure 12

Following the identification of the top 50 genes associated with each seed gene (*cpxA, ybjI,* and *thyA*), we conducted pathway enrichment analysis on the combined gene sets derived from these three seed genes. This analysis revealed that the genes were significantly enriched in two key pathways: the CAMP (Cationic Antimicrobial Peptide) resistance pathway and the Two-Component System pathway.

These pathways are known to play crucial roles in bacterial responses to environmental stresses, including antibiotic exposure. The CAMP resistance pathway is involved in mechanisms that allow bacteria to withstand the antimicrobial effects of cationic peptides, which are part of the innate immune system's defense against microbial pathogens. On the other hand, the Two-Component System pathway represents a fundamental regulatory mechanism in bacteria, mediating responses to various external stimuli, including antibiotic stress.

By pinpointing these pathways as significantly enriched among the genes associated with the seed genes, our analysis underscores their importance in bacterial adaptation and resistance mechanisms. Understanding the molecular processes underlying these pathways provides valuable insights into the strategies employed by bacteria to survive and thrive in challenging environments, such as those created by antibiotic exposure.

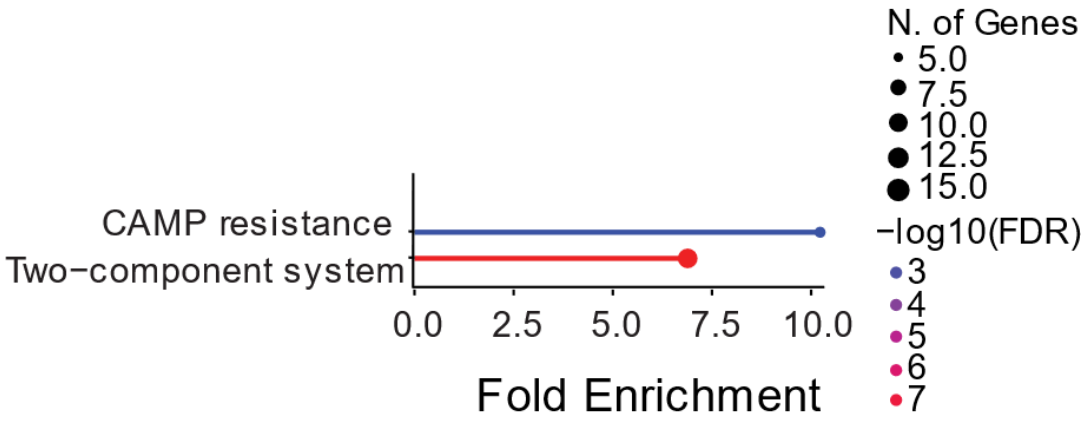


Figure 13

4) Gene analysis and LLR score

Results of carry forward and non carry forward gene

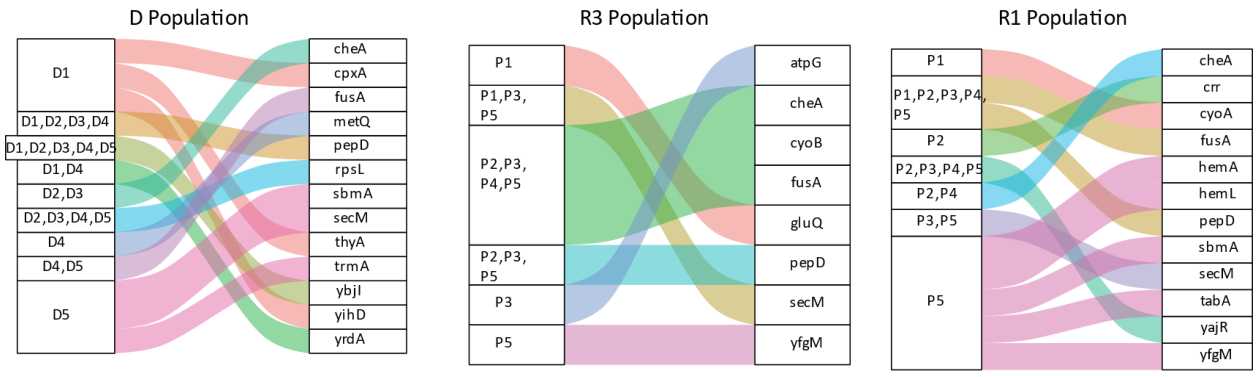


Figure 14

Analyzing carry-forward genes, which consistently appear in every passage under antibiotic pressure, and comparing them between D and R provides insights into critical resistance mechanisms, enabling the identification of potential intervention targets and informing the development of more effective antimicrobial strategies. Non-carry-forward genes, which do not appear consistently, may contribute to resistance more transiently, offering additional context for understanding differences in antibiotic resistance in D and R

LLR score result

In our analysis, we classified genes into two categories based on their association with antibiotic resistance: carry forward (CF) and non-carry forward (NCF). Subsequently, we assessed the likelihood of these genes being deleterious or pathogenic in nature using a log-likelihood ratio (LLR) score.

For the D population, the NCF genes exhibited a notably higher LLR score of -29.82, suggesting a greater likelihood of deleterious or pathogenic effects. Conversely, CF genes showed a lower LLR score of -7.3, indicating a more neutral or benign nature.

In contrast, for the R population, CF genes had an LLR score of -10.23, indicating a higher likelihood of being deleterious or pathogenic. Meanwhile, NCF genes displayed an LLR score of -9.47, suggesting a neutral or benign nature.

Notably, the seed genes, with LLR scores ranging from -3 to -8, were less disruptive to protein structure and function. Consequently, mutations within these genes were more likely to have a benign or neutral effect on protein function.

These findings provide valuable insights into the potential impact of genetic mutations on bacterial adaptation and antibiotic resistance. Understanding the nature of these mutations is essential for predicting their effects on bacterial fitness and informing strategies for combating antibiotic resistance effectively.

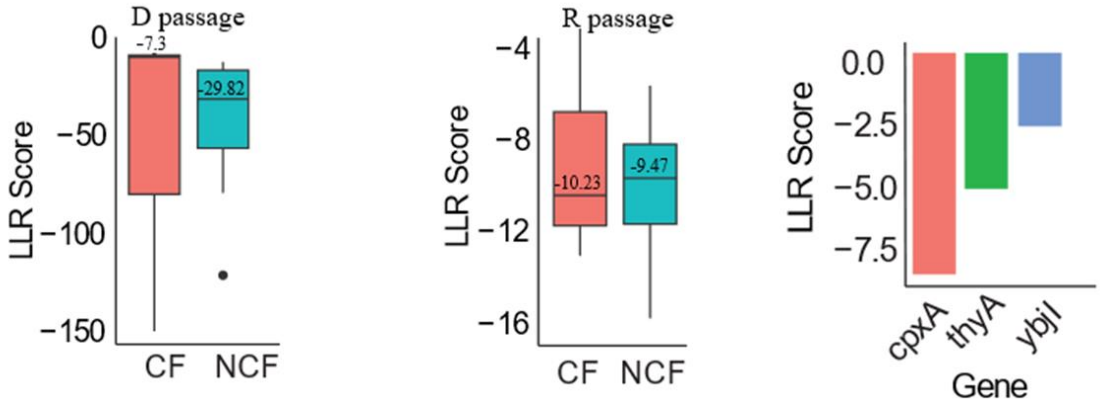


Figure 15

Chapter 6

Hypothesis

Through our comprehensive experiment, we have formulated a hypothesis concerning the mechanisms underlying antibiotic resistance in bacteria. Our approach involved employing Random Walk with Restart (RWR) to identify seed genes, followed by pathway enrichment analysis to elucidate the pathways associated with these genes. Notably, we found significant enrichment in the CAMP (Cationic Antimicrobial Peptide) pathway and the Two-Component System pathway.

In the context of the CAMP pathway, it is known that bacteria can produce cations that accumulate on their membranes, thereby hindering the entry of cationic antimicrobial peptides. Drawing on this understanding, we hypothesize that activation of the CAMP pathway may lead to an increase in cations on the bacterial membrane, preventing the entry of cationic antibiotics such as kanamycin. This mechanism could contribute to bacterial resistance against kanamycin.

Moreover, our investigation into the Two-Component System pathway revealed its activation under stress conditions, such as antibiotic exposure. Notably, previous research has shown that this pathway plays a role in bacterial responses to antibiotics like gentamicin[31]. Given the similar mechanisms of action shared by gentamicin and kanamycin as aminoglycoside antibiotics, we propose that activation of the Two-Component System in the presence of kanamycin may serve as a defense mechanism against the antibiotic.

By integrating these findings, we have gained insights into the pathways involved in bacterial resistance to kanamycin. Specifically, the activation of the CAMP pathway may prevent the entry of kanamycin into bacterial cells, while the

activation of the Two-Component System pathway may contribute to bacterial survival in the presence of kanamycin.

This hypothesis provides a framework for further experimental investigations into the molecular mechanisms of antibiotic resistance and may inform the development of novel strategies to combat antibiotic-resistant bacteria.

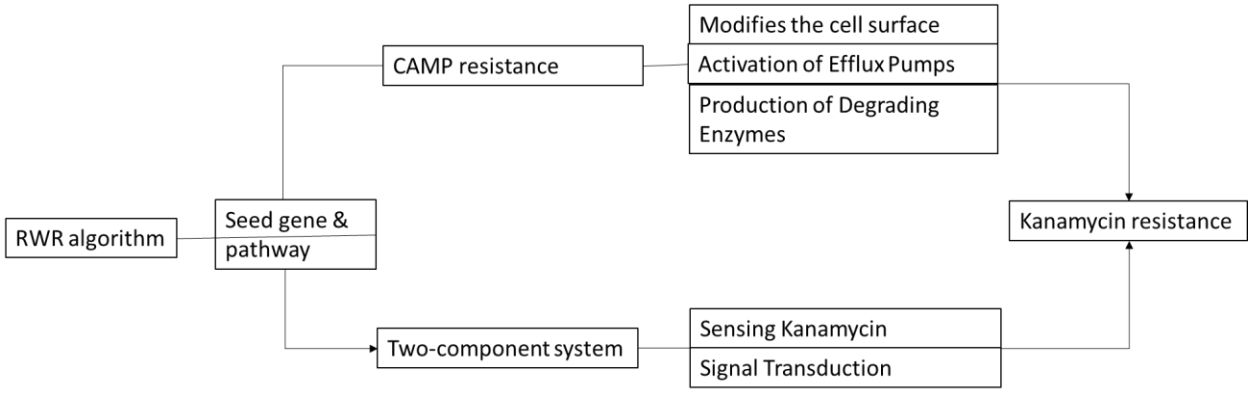


Figure 16

Chapter 7

Conclusion

In conclusion, through such approaches as those related to growth curve analysis or even gene knockout validation our comprehensive and multifaceted experimental approach has offered multiple insights into the adaptation mechanisms in bacteria under antibiotic stress. Pathway enrichment analyses, network analyses and careful sequencing enabled us to identify CAMP and Two-Component System pathways key genes-ybjI, thyA, cpxA.

Our approach integrates many types of analysis and biological methods to come up with a robust method for the identification of the genetic determinants of resistance to antibiotics. Our results indicate that the activation of the CAMP pathway might counteract the entry of some cationic antibiotics like kanamycin and more generally the conjunction between antibiotic defenses and two component system pathways.

The suggested gene knockouts serve as important validation steps to further confirm that these genes have an important role in bacterial resistance. The success of these experiments will reinforce our methodology and underscore the biological importance of these pathways for bacterial responses towards antibiotic-induced stresses.

Our research helps to understand the complexity of bacterial adaptation mechanisms and highlighting the importance of multi-disciplinary approaches in understanding the mechanism of antibiotic resistance. By shedding light on the molecular foundation of bacterial responses to antibiotics, our study contributes to the ongoing efforts to combat antibiotic-resistant infections and reduce the global health threat caused by antimicrobial resistance.

Chapter 8

Future plan

Gene knockouts are pending to be performed to illustrate our methodological approach. This is a verification exercise aimed at establishing that our method is efficient and reliable in the identification of leading genes involved in the bacterial resistance to antibiotics. We will focus on three important genes namely cpxA, ybjI and thyA which were extracted from an inclusive experimental pipeline; growth curve analysis, sequencing, Random Walk with Restart (RWR), pathway enrichment analysis, log-likelihood ratio (LLR) scoring and literature review.

The gene knockout experiments will involve targeted disruptions or deletions of cpxA, ybjI and thyA genes from the wild type strain of bacteria. After that we shall expose these knockout strains to similar concentrations of antibiotic as those used in our previous experiments. These knockout strains will be grown under antibiotic stress conditions so as to observe their growth phenotype.

If the knockout strains can grow under antibiotic stress like the wild-type strain did that would mean that our method was right because it has been validated by this outcome. Such observations may imply that these identified genes do indeed contribute significantly towards conferring antibiotic resistance to bacteria.
On the contrary, the wild type strain would have even more evidence to support these genes' importance if knockout strains show stunted growth or higher vulnerability in the face of antibiotic stress.

This is a very important step where we can confirm our methodology and if it is accurate enough or not since it identifies some of the major genes related to antimicrobial resistance. It also gives experimental proof that makes sense of biological implications behind pathways and mechanisms which were determined through our approach.

The aim of this research is to validate our approach by gene knockouts, thereby providing a stronger platform for our findings as well as contributing new knowledge on bacterial adaptation against antibiotic stress. Finally, such validation will further warrant using this method in future investigations of antibiotic resistance as well as bacterial adaptation.

Chapter 9

References

1) Barabási, A. L., & Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. Nature Reviews Genetics, 5(2), 101-113. doi:10.1038/nrg1272

2) Vidal, M., Cusick, M. E., & Barabási, A. L. (2011). Interactome networks and human disease. Cell, 144(6), 986-998. doi:10.1016/j.cell.2011.02.016

3) Graph Theory and Network Algorithms in Biology. Newman, M. (2018). Networks. Oxford University Press.

4) Albert, R. (2005). Scale-free networks in cell biology. Journal of Cell Science, 118(21), 4947-4957. doi:10.1242/jcs.02714

5) Kitano, H. (2002). Systems biology: A brief overview. Science, 295(5560), 1662-1664. doi:10.1126/science.1069492

6) Barabási, A. L., Gulbahce, N., & Loscalzo, J. (2011). Network medicine: a network-based approach to human disease. Nature Reviews Genetics, 12(1), 56-68. doi:10.1038/nrg2918

7) Cho, D. Y., Kim, Y. A., & Przytycka, T. M. (2012). Chapter 5: Network biology approach to complex diseases. PLoS Computational Biology, 8(12), e1002820. doi:10.1371/journal.pcbi.1002820

8) Zotenko, E., Mestre, J., O'Leary, D. P., & Przytycka, T. M. (2008). Why do hubs in protein-protein interaction networks tend to be essential: reexamining the connection between the network topology and essentiality. PLoS Computational Biology, 4(8), e1000140. doi:10.1371/journal.pcbi.1000140

9) Barabási, A. L., & Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. Nature Reviews Genetics, 5(2), 101-113.

10) Köhler, S., Bauer, S., Horn, D., & Robinson, P. N. (2008). Walking the interactome for prioritization of candidate disease genes. American Journal of Human Genetics, 82(4), 949-958.

11) Alon, U. (2007). Network motifs: theory and experimental approaches. Nature Reviews Genetics, 8(6), 450-461.
12) Cowen, L., Ideker, T., Raphael, B. J., & Sharan, R. (2017). Network propagation: A universal amplifier of genetic associations. Nature Reviews Genetics, 18(9), 551-562.
13) Yildirim, M. A., Goh, K. I., Cusick, M. E., Barabási, A. L., & Vidal, M. (2007). Drug-target network. Nature Biotechnology, 25(10), 1119-1126.
14) Valdeolivas, A., Tichit, L., Navarro, C., Perrin, S., Odelin, G., Levy, N., ... & Baudet, L. (2019). Random walk with restart on multiplex and heterogeneous biological networks. Bioinformatics, 35(3), 497-505.
15) Lee, D. S., Park, J., Kay, K. A., Christakis, N. A., Oltvai, Z. N., & Barabási, A. L. (2008). The implications of human metabolic network topology for disease comorbidity. Proceedings of the National Academy of Sciences, 105(29), 9880-9885.
16) Brandes, N., Goldman, G., Wang, C.H. et al. Genome-wide prediction of disease variant effects with a deep protein language model. Nat Genet 55, 1512–1522 (2023).
17) Sambrook, J., Fritsch, E. F., & Maniatis, T. (1989). Molecular Cloning: A Laboratory Manual (2nd ed.). Cold Spring Harbor Laboratory Press.
18) Green, M. R., & Sambrook, J. (2012). Molecular Cloning: A Laboratory Manual (4th ed.). Cold Spring Harbor Laboratory Press.
19) Sezonov, G., Joseleau-Petit, D., & D'Ari, R. (2007). Escherichia coli Physiology in Luria-Bertani Broth. Journal of Bacteriology, 189(23), 8746-8749.
20) Marciano, D.C., Wang, C., Hsu, TK. et al. Evolutionary action of mutations reveals antimicrobial resistance genes in Escherichia coli. Nat Commun 13, 3189 (2022).
21) Hice, S.A., Santoscoy, M.C., Soupir, M.L. et al. Distinguishing between metabolically active and dormant bacteria on paper. Appl Microbiol Biotechnol 102, 367–375 (2018)
22) Andrews S. FastQC A Quality Control tool for High Throughput Sequence Data [Internet].
http://www.bioinformatics.babraham.ac.uk/projects/fastqc/. [cited 2017 May 3]. Available from:

http://www.bioinformatics.babraham.ac.uk/projects/ fastqc/.
23) MultiQC: summarize analysis results for multiple tools and samples in a single report |
Bioinformatics | Oxford Academic. (n.d.). Retrieved June 14, 2017, from
https://academic.oup.com/bioinformatics/article/32/19/3047/2196507/MultiQC-summarize-analys
is-results-for-multiple.
24) Shifu Chen, Yanqing Zhou, Yaru Chen, Jia Gu; fastp: an ultra-fast all-in-one FASTQ preprocessor,
Bioinformatics, Volume 34, Issue 17, 1 September 2018, Pages i884–i890,
https://doi.org/10.1093/bioinformatics/bty560.
25) Li H, Durbin R. Fast and accurate short read alignment with Burrows–Wheeler transform.
Bioinformatics. 2009;25:1754–60.
26) “Picard Toolkit.” 2019. Broad Institute, GitHub Repository. https://broadinstitute.github.io/picard/;
Broad Institute
27) Tarasov A, Vilella AJ, Cuppen E, Nijman IJ, Prins P. Sambamba: fast processing of NGS alignment
formats. Bioinformatics. 2015;31:2032–4.
28) Garrison E, Marth G. Haplotype-based variant detection from short-read sequencing.
arXiv:12073907 [q-bio]. 2012. http://arxiv.org/abs/1207.3907. Accessed 24 May 2017.
29) Cingolani P, Platts A, Wang LL, Coon M, Nguyen T, Wang L, et al. A program for annotating and
predicting the effects of single nucleotide polymorphisms, SnpEff. Fly (Austin). 2012;6:80–92.
30) "Using Drosophila melanogaster as a model for genotoxic chemical mutational studies with a new
program, SnpSift", Cingolani, P., et. al., Frontiers in Genetics, 3, 2012.

31) Ćudić, E., Surmann, K., Panasia, G. et al. The role of the two-component systems Cpx and Arc in protein alterations upon gentamicin treatment in Escherichia coli . BMC Microbiol 17, 197 (2017).