Plasmodium Falciparum Microarray Data Analysis Using Machine Learning Approaches

By: Abdiwak Tesema Tujo

A Thesis Submitted to the Department of Computing

School of Electrical Engineering and Computing.

Presented in Partial Fulfillment for the Degree of Master's of Science in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

Adama, Ethiopia,

February 2022

Plasmodium Falciparum Microarray Data Analysis Using Machine Learning Approaches

By: Abdiwak Tesema Tujo

Advisor: Tilahun Melak (PhD.)

A Thesis Submitted to the Department of Computing

School of Electrical Engineering and Computing.

Presented in Partial Fulfillment for the Degree of Master's of Science in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

Adama, Ethiopia,

February 2022

Declaration

I hereby declare that this MSc Thesis is my original work and has not been presented for a degree in any other university, and all sources of material used for this thesis have been duly acknowledged.

Name: Abdiwak Tesema

Signature: _____

This MSc Thesis has been submitted for examination with my approval as thesis advisor.

Name: Tilahun Melak (PhD.)

Signature: _____

Date of submission: _____

Approval of Board of Examiners

We, the undersigned, members of the Board of Examiners of the final open defense by Abdiwak Tesema Tujo have read and evaluated his/her thesis entitled "Plasmodium Falciparum Microarray Data Analysis Using Machine Learning Approaches" and examined the candidate. This is, therefore, to certify that the thesis has been accepted in partial fulfillment of the requirement of the Degree of Master's of Computer Science and Engineering.

_____	_____	_____
Advisor	Signature	Date
_____	_____	_____
Chairperson	Signature	Date
_____	_____	_____
Internal Examiner	Signature	Date
_____	_____	_____
External Examiner	Signature	Date

Acknowledgment

First and above all, I would like to acknowledge the Holly Creator for his grace in giving me the knowledge, strength, and good health to accomplish my research. I would like to express my gratitude to Tilahun Melak (Dr.), my advisor for believing in my ability and allowing me to work on this project for my master's thesis. His deep understanding has enriched my research work. He is always being a source of support and motivation for bringing the quality work. He has been very supportive throughout the entire work.

I am very much indebted to the Artificial Intelligence SIG members of the Computer Science and Engineering department students and staffs for their valuable suggestions in my research work. I am very thankful to all teachers and staff members of the department for their support, advice, and encouragement during my study time. I also acknowledge for the resources that I have received from the department, AI SIG, and from the University.

Finally, yet most importantly, I am very much grateful to my family, my mother Mrs. Zekiyat Helil, my father Mr. Tesema Tujo, my wife Mrs. Sualiha Gebeyaw, my aunt Miss. Rukiya Helil, my uncle Mr. Neja Helil, and my cousin Miss. Misra Neja. I am very lucky and blessed to have you in my life. Thanks for believing in me, supporting me, understanding me, and for your patience in all my efforts.

Table of Contents

List of Figures

List of Tables

List of Equations

List of Algorithms

List of Abbreviations and Acronyms

DEG	Differentially Expressed Gene
DE	Differential Expression
DNA	Deoxyribonucleic Acid
LIMMA	LInear Models for MicroArray
ML	Machine Learning
P. falciparum	Plasmodium Falciparum
RNA	Ribonucleic Acid
SVM	Support Vector Machine
WHO	World Health Organization
MA	Microarray Data
UML	Unified Modelling Language
ERD	Entity Relation Diagram

Abstract

Malaria is one of the deadliest diseases to humans. The disease is developing resistance to antimalarial drugs in different countries worldwide. In addition to this problem, there is no vaccine available despite decades of research. The systematic development of resistance to antimalarial drugs forces researchers to generate a massive volume of data. The data retrieved from Microarrays shows this fact. This research objective is to identify drug targets at the gene level from gene expression data obtained from microarrays. In this research we perform the preprocessing of the raw MA Data before high-level analysis. After loading the raw data into the working environment of the R studio, first we explore the dataset to check if it contains the required components. Then we apply microarray data quality control and after that we go for background correction, normalization, and log2 transformation of the data. Finally, we check for the existence of missing values and screening of outliers. We applied the Empirical Bayes method to identify differentially expressed genes in the high-level analysis. Identification of differentially expressed genes results in 2500 differentially expressed genes out of 22769 genes. The study applied clustering of DEGs to group them based on their expression values, considering that genes within the same cluster have the same biological behavior. We applied the hierarchical clustering technique. The clustering result gives us 226 genes falling in the fourth cluster, 283 genes in the third cluster, 616 genes in the first cluster, and 904 genes in the second cluster. We validated the clustering result using an internal validation measure in which hierarchical clustering is selected as the best clustering technique for this study. The final step in this research is constructing a gene-gene interaction network for the up-regulated genes. We used Cytoscape software to construct the network. Finally in this study, from the network we extracted top 20 genes that can be used as drug targets.

Keywords: - *Gene, Gene Expression, Malaria, Plasmodium Falciparum, Microarray Data, Machine Learning, Clustering, Gene-gene Interaction, Differential Expression*

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

Malaria has been a life-taking disease to humans from ancient times until nowadays. Records from the Chines document about 2700 BC ago, in Mesopotamia's clay tablets before 2000 BC, the reports on papyri of Egyptians from 1570 BC, and Hindus texts recorded back in the sixth century BC indicated that the disease was in existence during the ancient times[1]. Those records described it as a disease with the characteristics of intermittent fever on an infected person. Plasmodium is a unicellular parasite that commonly causes disease, and the bite from an infected Anemophilous Gambie mosquito can transmit the disease. There are more than 100 distinctive types of parasite Plasmodium. Among those, only five of them are causatives to human Malaria disease, including Plasmodium falciparum, malariae, vivax, knowlesi, and ovale[2]. The falciparum family of Plasmodium parasites is prevalent worldwide among the five recognized species [2], [3].

In 2017, K. N. Olafson, Tam Q. Nguyen, et al., [4] approximately 3.2 billion people lived in malaria-endemic areas in Africa, South America, and South-East Asia. In the 2019 malaria report of the WHO, the Plasmodium falciparum parasite was responsible for 99.7 percent, 50 percent, 71 percent, and 65 percent of the issues in the World Health Organization Africa Region, WHO South-East Asia Region, WHO Eastern Mediterranean Region, and WHO Western Pacific Region, respectively. The recent malaria report[5] shows 241 million malaria issues in 2020, up from 227 million in 2019.

The death number in 2020 is 627,000, a rise of 69,000 deaths compared to 2019[5]. From this number, 47,000 were due to the interruptions of the COVID-19 pandemic. The remaining 22,000 represent a recent modification in WHO's approach for determining the figure of malaria-related deaths (regardless of COVID-19 disruptions). The report concludes that Plasmodium Falciparum infection is the world's most pressing humanitarian challenge. The WHO report indicates that about half of the world's population is in danger of being infected. The world must eradicate malaria before it causes further damage to people.

According to WHO [6], eradicating malaria is a major global and public health priority for decades. It makes several recommendations for malaria prevention. It proposes various malaria prevention methods. The first method is vector control, which assists in preventing and reducing malaria transmission; the second method suggested by the WHO is treatment with an antimalarial drug. Even though using antimalarial drugs for treatment is one of the controlling mechanisms, the Plasmodium parasite develops resistance to the treatments.

The Parasites of Plasmodium are genetically diverse. As the parasites go through the mosquito, they mate and exchange their pieces of DNA with a human host. According to Q. Jing et al. [7], the parasite can replicate itself within the human red blood cell, causing disease. They also revealed that a single P. Falciparum parasite contains ~60 genes. The var(disease-causing) gene of the parasite can differ in expression (differential expression) at a point in time during replication.

According to [8],[9],[10],[11],[12] we can learn a great deal about an organism's gene expression by analyzing its gene expression data. Studying the biology/genome of the P. falciparum parasite's biology/genome is the solution to deal with the drug resistance problems mentioned above. DNA microarray technologies can help researchers better understand the biology of parasites in various physiological conditions or states[13]–[20]. We can learn about biological processes by analyzing the massive amounts of data generated by microarray experiments using machine learning techniques.

Computer scientists, biologists, statisticians, data experts, and other specialists analyze rapidly changing problems due to technologies and new genomics experiments. The technology makes it possible to look at organisms in a genome-wide manner. One of the new kinds of experiments is the high throughput experiment, and the massive amount of biological data gotten from these experiments makes it different from the earlier experiments. Using the technology generically known as "Microarrays," it is possible to explore the functional behavior of an organism's genes under diverse conditions or experiments using only one experiment. These experiments can generate data with thousands to millions of variables [21].

In light of this, we proposed this study to use machine learning to analyze the massive amount of biological data. This study analyzed raw DNA microarray data from the

Plasmodium falciparum parasite, which is publicly available. In this study, we used the unsupervised technique of machine learning algorithms. We analyzed the data using the R programming language. We used the Bioconductor packages in R, starting from the preprocessing stage until the final analysis. We got a list of DEGs from the study results, and we applied ML algorithms to those DEGs. For identifying DEGs we used limma (Linear Model foe Microarray Analysis). Those differentially expressed genes are the most significant/ informative genes. Hierarchical clustering algorithm of the machine learning technique is used to cluster genes with the same biological behavior.

1.2. The motivation of the study

The disease malaria is fatal to the whole human race. Children, pregnant women, elder peoples, as well as those young ones suffers from the disease. The threat remains a global issue with resistance to antimalaria drugs, and there is no vaccine available despite decades of research [22]. The WHO's latest malaria report[5] of 2021 shows that in the year 2020, there were 241 million cases of malaria and 227 million cases in 2019. Plasmodium parasites can carry the disease from the infected person's blood to uninfected person during their mealtime of blood. There are five Plasmodium parasite families, and the falciparum family is the most public of them all. A complete understanding of P. falciparum's molecular biology will be vital for developing new drugs and vaccine development strategies to deal with its antimalarial drug resistance.

In order to eradicate this deadliest disease, I thought that I am one of the responsible persons. The other thing that motivates me to do the research is working with big data. Lastly but not the least I am very interested in Bioinformatics concepts. I want to specialize in Bioinformatics in the near future and I see this research as a footstep for me. Since we have many P. falciparum genomic data, this study applies the different ML approaches to study the parasite's functional behavior. As part of our study, we used ML to analyze the parasite's gene expression.

1.3. Statement of the Problem

Life-threatening disease: Malaria can characteristically spread through the bite of an infected Anopheles Gambie mosquito, which carries the Plasmodium parasite. When this mosquito bites the host organism, it releases the parasite into the host's bloodstream. Only 30-40

species of Anopheles Gambie mosquitos can transmit the genus Plasmodium's parasite out of the 460 identified species of Anopheles[23]. Plasmodium Parasites are genetically diverse. As the parasites pass through the mosquito, they mate and exchange their pieces of DNA with a human host. The parasite replicates itself within the human red blood cell. During this time, it causes the disease malaria.

To get relief, an infected person can take medicines. Those medicines are typically small molecules that bind to a particular protein in the body which can inhibit or activate its activity—however, P. falciparum develops resistance to antimalarials. Drug resistance endangers global efforts to control and eradicate the disease.

Genomic data analysis in the parasite's stage-specific gene expression is essential for drug and vaccine production. We need to analyze the differentially expressed gene to deal with the P. falciparum parasite's drug resistance. We can explicitly analyze the DEG from the abundant MA Data using ML techniques. Microarray technologies generate enormous amounts of data, and analyzing this data is one of the foremost challenges to effectively using microarray technology. Gene expression studies that use microarrays always start with a biological question. Following an idea of Yongliang Yang et al.[24] we used machine learning techniques to identify potential drug targets by constructing a gene-gene interaction network of P. falciparum using a microarray dataset.

Using stage-specific metabolic network analysis, Neel Dholakia, Pinakin Dhandhukia, *et al*. [25] identified potential target of Plasmodium falciparum using MA data. In the differential expression analysis, they used ANOVA. Under tight assumptions about the nature of the microarray data, it is impossible to do an ANOVA analysis properly. The fact that there is no unique interpretation of the significance of two means of the genes makes it less useful than the t-test, which is a more powerful test. Further testing is necessitated by the requirement of the post-ANOVA t test. Using microarray technology, Massoume Eskandari, Shahla Chaichian, *et al.*[26] studied ovarian cancer microarray data in order to discover a pattern for detecting gene expression alterations in ovarian cancer. Due to the fact that they did not validate their clustering results, their study overlooks the critical difficulties that are critical to the success of the cluster analysis. The validation process can be used to determine the quality of the clustering result.

The existence of vast amount of biological data for the identification of drug targets tackles various challenges such as handling the data from the microarray experiments, the right use of the available analysis technique, and getting the targets. This study addressed the issue related to drug target identification by answering the following questions using MA Data from the most prevalent P. falciparum strain and machine learning techniques:

- → How to identify Differentially Expressed Plasmodium Falciparum Anopheles Gambie mosquito genes?
- → How to identify Plasmodium Falciparum Anopheles Gambie genes with the same biological behavior?
- → Which of the Plasmodium Falciparum Anopheles Gambie genes are potential drug targets?

1.4. Objectives of the Study

This study applies existing linear model for microarray analysis, and machine learning algorithms to analyze the P. falciparum microarray data. The ultimate contribution of the study was to identify genes that can be used as potential drug targets using gene expression analysis models.

1.4.1. General Objective

This research's general objective was to analyze Plasmodium falciparum's gene expression MA Data using machine learning approach to identify potential drug targets.

1.4.2. Specific Objective

This research's specific objectives were to:

- → To identify differentially expressed genes.
- → To cluster genes with similar expression profiles together.
- → To validate the clustering model.
- → To identify genes that are potential drug targets.

1.5. Data Collection

Data Collection is a fundamental aspect of any analysis study, and the data collection starts with determining what kind of data are needed to be followed by data collection techniques.

We extracted Plasmodium falciparum's MA Data from a publicly available repository, and we extracted the data from the National Center for Bioinformatics website. We used raw MA Data of P. falciparum in this study.

1.6. Significance of the Study

This study's contribution was to identify potential drug and vaccine targets for malaria disease caused by P. falciparum species. Plasmodium falciparum's vast MA Data of Plasmodium falciparum's gene expression data was analyzed to identify the most informative genes. Machine learning techniques are applied to study genes with the same biological behavior and identify the potential targets using a gene-gene interaction network. This study's primary beneficiaries are those who can develop drugs and vaccines for the disease of malaria. Countries where malaria is prevalent can be the beneficiaries of this study too.

1.7. Scope and Limitations of the Study

1.7.1. Scope of the Study

Five malaria parasite species can infect humans; Plasmodium vivax, Plasmodium ovale, Plasmodium malariae, Plasmodium knowlesi, and Plasmodium falciparum[2]. We studied the gene expression analysis for P. falciparum MA Data only. There are many genomic data for analyzing Plasmodium falciparum's gene expression, RNAseq, protein microarray, and data from various literature and texts. However, in this study, we used gene MA Data only. This study used only unsupervised machine learning approaches, excluding supervised and semi-supervised techniques.

1.7.2. Limitations of the Study

This study comes across limitations on a different phase of the research development process. This study's discussion focused mainly on P. falciparum MA Data analysis, assuming that the data was appropriate and generated with careful experimental design. Hence in this study, we did not perform the MA Data generation experiments due to a lack of lab resources to experiment in the lab. The other limitation of this study is that we did not perform the PRISMA test for the systematic literature review. The last limitation of the study is that it does not find a solution for the significant overlapping of genes between groups.

1.8. Application of Results

The study extends research related to Plasmodium falciparum's drug resistance. The research contributed its part in the identification of potential drug targets. The benefits of the research were:

- → The identified genes can be used as candidate genes and help prioritize those genes.
- → Pharmacologists can use the study result to develop potential drugs or vaccines.
- → The study results combined with analyzing text or literature data used for integrated data mining.
- → It will save the amount of money funded needed for malaria research.
- → It helps to understand the essential biological function of the P. falciparum gene.
- → To identify essential regulatory genes involved in the biological process.

1.9. Operational Definition of Terms

1.9.1. Genome

A genome is either a complete set of genetic information or the entire genetic material set in an organism's cell[27]. Malcolm J. Gard et al.t al[28]; stated that the P.falciparum3D7 strain's nuclear genome comprises 22.8 mega-base(Mb) distributed among the 14 chromosomes P. falciparum varying in size from 0.643 to 3.29Mb approximately.

1.9.2. Proteome

The proteome is the whole set of protein products expressed by the genome. Due to protein-protein interactions in the cell's environmental conditions and becomes an unstable entity.

1.9.3. DNA

DNA is a Deoxyribose Nucleic Acid that encodes genetic information in organisms. DNA can replicate itself and synthesize RNA. There are four kinds of nucleotides in DNA; Adenine(A), Guanine(G), Thymine(T), and Cytosine(C). The four bases of the alphabet can encode genetic information. The letters are read-only in one direction.

1.9.4. RNA

RNA is a nucleic acid molecule containing the sugar component ribose. The base pairs of RNA are different from that of DNA in which the uracil (U) base is instead of thymine (T) in DNA. RNA has much shorter sequences of nucleotides than DNA[27].

1.9.5. Gene

Gene is the DNA segment transcribed into RNA, which may then translate into a protein. Proteins or RNA transcripts are functional products encoded by genes. Each gene is associated with a promoter sequence that initiates gene transcription and regulates its expression[27].

1.9.6. Gene Expression and Gene Expression Level

Gene expression is defined as the process by which information contained in a gene is converted into a functioning product. In contrast, the gene expression level is the number of copies of RNA transcripts created by a particular transcription gene[27]. Expressed genes include protein-coding genes and genes coding for RNA functional products. Most of the time, the gene expression focuses on the expression level of protein-coding genes[27]. Expression of genes results in protein synthesis encoded by the gene within the cell. The higher the expression level is, the more protein production from the expression

1.9.7. Protein

Protein is a biological macromolecule constituted by one or many amino acid chains. DNA sequences of the genes coding for the protein direct the protein production[27]. Proteins are fundamental modules of all living cells, and each of them has exclusive functions. P. falciparum can infect erythrocyte membrane protein 1, a family of proteins present on the host's membrane surface of red blood cells[29].

1.10. Organization of the Thesis

The research paper is organized into seven chapters. The first chapter describes the general research area, including the study's background, motivation, critical terms related to domain knowledge, problem statement, objectives of the study, data collection methods, the significance of the study, scope and limitation of the study. The second chapter covers a

detailed literature review and related works on P. Falciparum Microarray Data analysis, topics related to malaria, detailed information about microarray, application of machine learning in microarray data analysis, machine learning approaches, and gene-gene interaction network. Chapter three is about the research methodology used in this research paper. The *fourth chapter* presents the proposed solution for the P. Falciparum MA Data analysis. It contains the diagrammatic and detailed representation conceptual framework of the study, preprocessing, and high-level analysis. The fifth chapter discusses the implementation and experimentation of the proposed solution. This chapter also presents the implementation and deployment environment of the study. *Chapter Six* discusses the results for preprocessing and high-level implementation. It also discusses the results for evaluation metrics for clustering. Finally, *Chapter Seven*; stipulates the conclusion of this research study, recommendation, and future works of the study.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORKS

In this chapter, we reviewed related literature to add more understanding of the problem background. It shows what has already been done by other researchers, which strongly relates to the present study. Literature related to the subject matter, methodology, and design of the study, is reviewed. It started by discussing malaria, Machine Learning Approaches, MA Data, P. falciparum MA Data, Applications of Machine Learning in MA Data, Clustering, Entity Filtering, and finally, Related Works.

2.1. Malaria

Malaria is among the deadliest diseases, and it has threatened humans for centuries. The Plasmodium parasite genus can cause malaria disease [1], [30]–[34]. There are five families of the parasitic protozoan causing malaria in humans; Plasmodium Falciparum, Plasmodium Vivax, Plasmodium Malariae, Plasmodium Knowlesi, and Plasmodium Ovale[2]. Infected mosquitos of Anopheles Gambiae insect can transmit the disease because of the parasite's genus, which they can carry within their body. An infected Anopheles Gambiae can release the parasite into humans' bloodstream or other host animals during their blood meal.

2.1.1. History of Malaria

Knowing where the history of malaria starts is essential. According to Virander S. Chauhan1 *et al.* [17], the earliest medical reports from early empires in Mesopotamia, China, and India have described malaria as a disease characterized by intermittent fevers. Their study notes that the Greek physician Hippocrates classified the disease according to the periodicity of every third day (ferbis tertian) and every fourth day (ferbis quartana). He also described its association with splenomegaly. The study also indicates that the name of the disease "Malaria" originates from the Medieval Italian Mal, "Bad," and Aria "Air" words, which the Romans of the Middle Ages originate. They understand that the illness's cause is toxic fumes from the swampy lands during that time. Malaria cases declined after the swampy lands' dryness, further revealing this belief during that time [31].

Alphonse Laveran (1845-1922), a military member and a physician based in Algeria, identified the disease's actual causative agent in 1880. He found a crescent-shaped malaria parasite in a soldier suffering from intermittent fever in the red blood sample. In 1881 he

reported an animal parasite after observing a motile filamentous structure that emerges from a round spherical body, and he named it Oscillaria Malariae. His findings became the foundation for further study of the disease[31].

V. S. Chauhan et al. [31] observed the difference between the two types of malaria in 1886. They also stated the morphological variations of the parasites. According to the report of Golgi, the parasite underwent asexual reproduction. He discovered that the lysis of the red blood cells and the parasites' release are the reason for the fever. According to the research, the parasite's two variant species were named Plasmodium Vivax and Plasmodium Malariae in 1890 by Grassi and Feletti.

The third type of the plasmodium parasite family, which is called Plasmodium falciparum, was identified by Sakharov (1889), Marchiafava (1890), and Celli (1890). In the year of 1890, the leading root of malaria, a protozoan parasite that invades and multiplies in red blood cells, was identified. According to their periodic specificities and other features, Benign tertian (Plasmodium Vivax), malignant tertian (Plasmodium Falciparum), and quartan malaria (Plasmodium Malariae) was discovered[31]. Among the parasites, P. falciparum is the deadliest, and: also, its prevalence is very high all over the world. The world has been suffering from this disease for an extended period.

2.2. What is The Cause of Severe Malaria Disease?

According to Ferhat Ay et al. [33], Cytoadherence causes severe disease and prevents the removal of abnormal RBCs and intraerythrocytic inclusions. The presence of virulence protein parasites on the surface of RBCs can facilitate cytoadherence. These are P. falciparum Erythrocytic Membrane Protein 1 (PfEMP1). Approximately there are 60 different PfEMP1 variants within a single P. falciparum parasite. The change in gene expression level is an essential factor for P. falciparum development in the different phases of its life cycle.

Virulence genes can encode different PfEMP1, which only one can express at a given point in time. Gene expression at the instant of time is an essential factor for the var gene to get away from the host's immune responses.

2.3. Plasmodium Falciparum

P. falciparum is one of the protozoan parasites that causes the most infectious human malaria and is also one of the deadliest parasites. Many morphological changes occur during the transmission of Anopheles Gambiae to humans[35]. The parasite disrupts the normal physiology of its host during the blood stage of its development. It grows and divides within the host's RBCs, feeding on the hemoglobin and remodeling its host cells to adhere to blood vessel walls[35].

2.3.1. The Complete Life Cycle of Plasmodium Falciparum Parasite

The above-stated research helps us understand the complete life cycle typical to all human Plasmodium parasite species reviewed in Figure 2.1 [36]. From the study of Ferhat Ay et al. [32], we can see that the P. falciparum parasite's complicated life cycle encompasses three major developmental stages: mosquitos, a human liver, and human blood. Virander S. Chauhan1 *et al.* [31] stated that the bite of an infected Anopheles gambiae starts the Plasmodium parasites' life cycle. In blood meal, the infected Anopheles Gambiae releases the spindle-shaped invasive into the skin and then moves to the bloodstream. This spindle-shaped invasive is known as Sporozoites.

According to [26], Sporozoites multiply rapidly in hepatocytes within two weeks. In this environment, a single Sporozoite can produce approximately forty thousand invasive structures known as Merozoites. They invade RBCs after releasing the new Merozoites into the bloodstream. Schizogony is how the parasite reproduces asexually within the red blood cells. Following asexual reproduction, a single parasite can produce between 16 and 32 daughter Merozoites and develops through three distinct developmental stages: Ring, Trophozoite, and Schizont. The release of daughter Merozoites into the bloodstream initiates a new invasion and growth cycle in the erythrocyte [31][32].

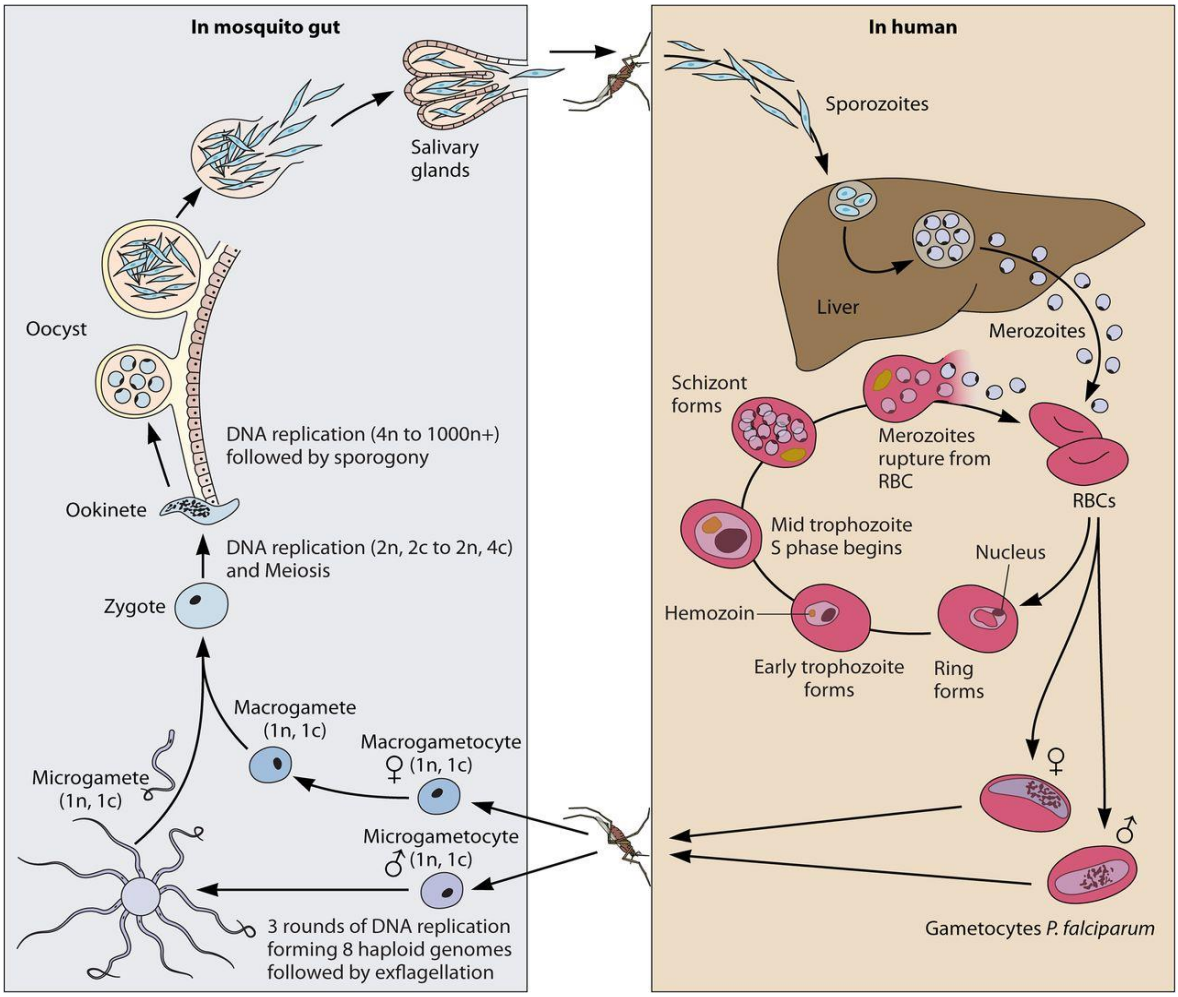


Figure 2.1. The Complete Life Cycle of Plasmodium Falciparum Parasite.

The Plasmodium Falciparum parasite life cycle can also include sexual development, resulting in male and female Gametocytes[32], [37]. After mosquitos feed on mature gametocytes, the parasite develops further within the mosquito's midgut [31][32].

2.4. The Genome of Plasmodium Falciparum Parasite s

According to [38], the genome of Plasmodium parasites is approximately 26 mega-bases. There is a distribution of genes on 14 chromosomes, and this genome encodes over 5000 genes. The expression of the P. falciparum gene is in a stage-specific manner. There is a difference between the expression levels of genes and gene products in the mosquito's midgut and the host's liver schizont.

2.5. Microarray Technology

David Botstein developed microarrays in the mid-1990s with Patrick O. Brown and Joseph DeRisi[39]. Large sets of annotated clones and sequences from high-throughput expressed sequence tags or genome sequencing projects can build microarrays. High throughput means many observations per sample, and this technology can create massive expression patterns of hundreds of thousands of genes simultaneously. Microarray data on gene expression is multidimensional and multifaceted, and its primary goal is to find genes that differ between samples or conditions[11].

The physical delivery system used to collect data contains many new and unknown functional relationships. Spots or features are specific locations where DNA molecules are orderly placed in a microarray. There may be millions of identical DNA molecules in each microarray spot representing a gene. DNA on a microarray can be genomic DNA or a gene-specific oligonucleotide strand (ONS)[40]. Figure 2.2 [41] depicts the schematic overview of probe array and target preparation for spotted cDNA microarrays and high-density.

oligonucleotide microarrays.

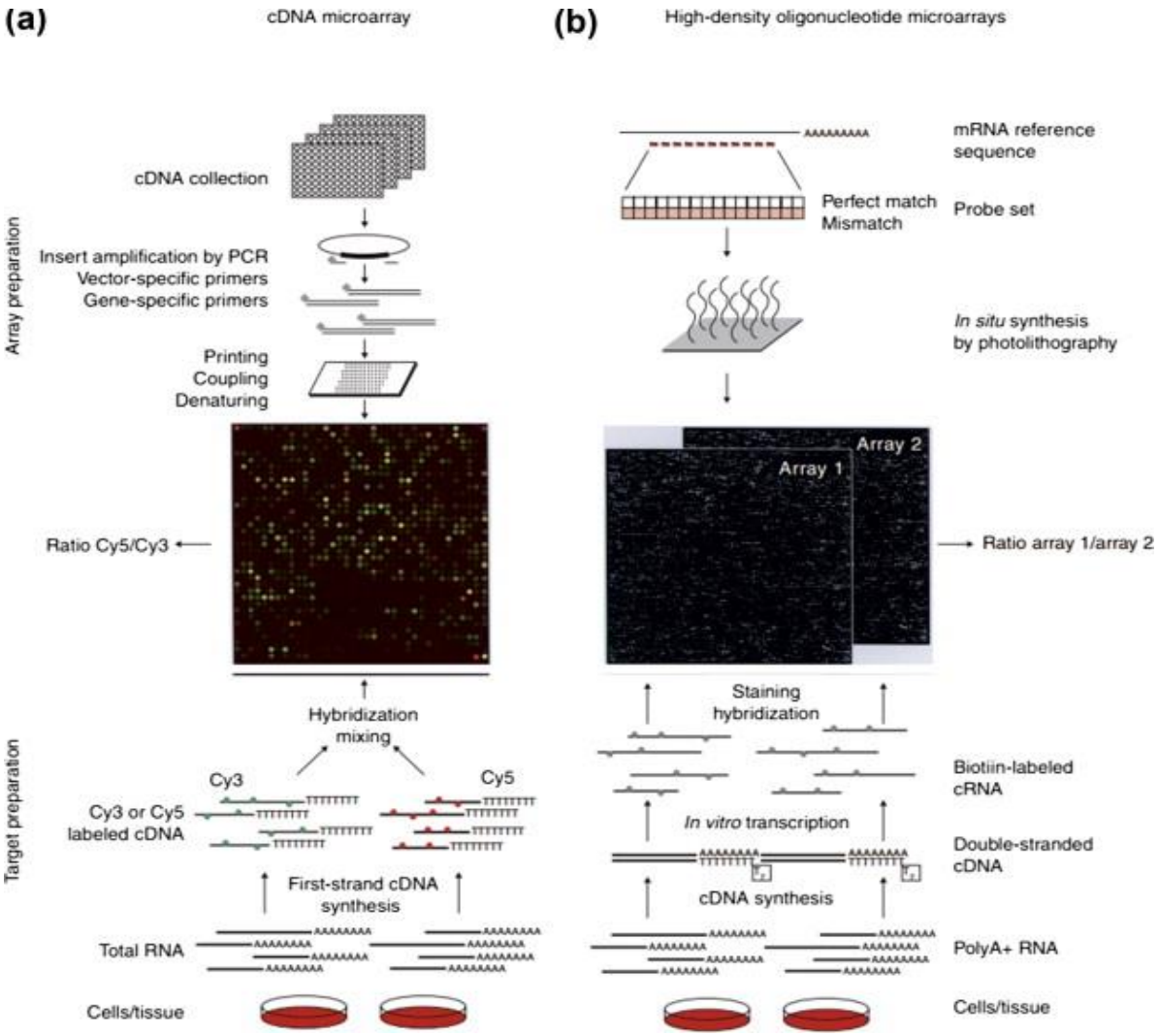


Figure 2.2. cDNA and High-density oligonucleotide Microarray

2.6. Microarray Data

We can now understand the molecular and gene revolution using microarray technology because we can simultaneously look at all genes [42]. Microarray technology provides data to measure gene expression levels and track changes in gene expression levels[43]. In MA Data matrices, rows represent genes, and columns represent different experimental conditions or samples. Each point in the data matrix can represent a particular gene's expression level for the given sample or experimental condition. The entries in a row or a column can form an expression pattern. In a typical gene expression matrix data set, there may be between 10 and 100 columns (different experimental conditions or samples) and between 10,000 and 100,000 rows (entities) (genes). The following Figure 1.3 [44] shows the generation and structure of MA Data.

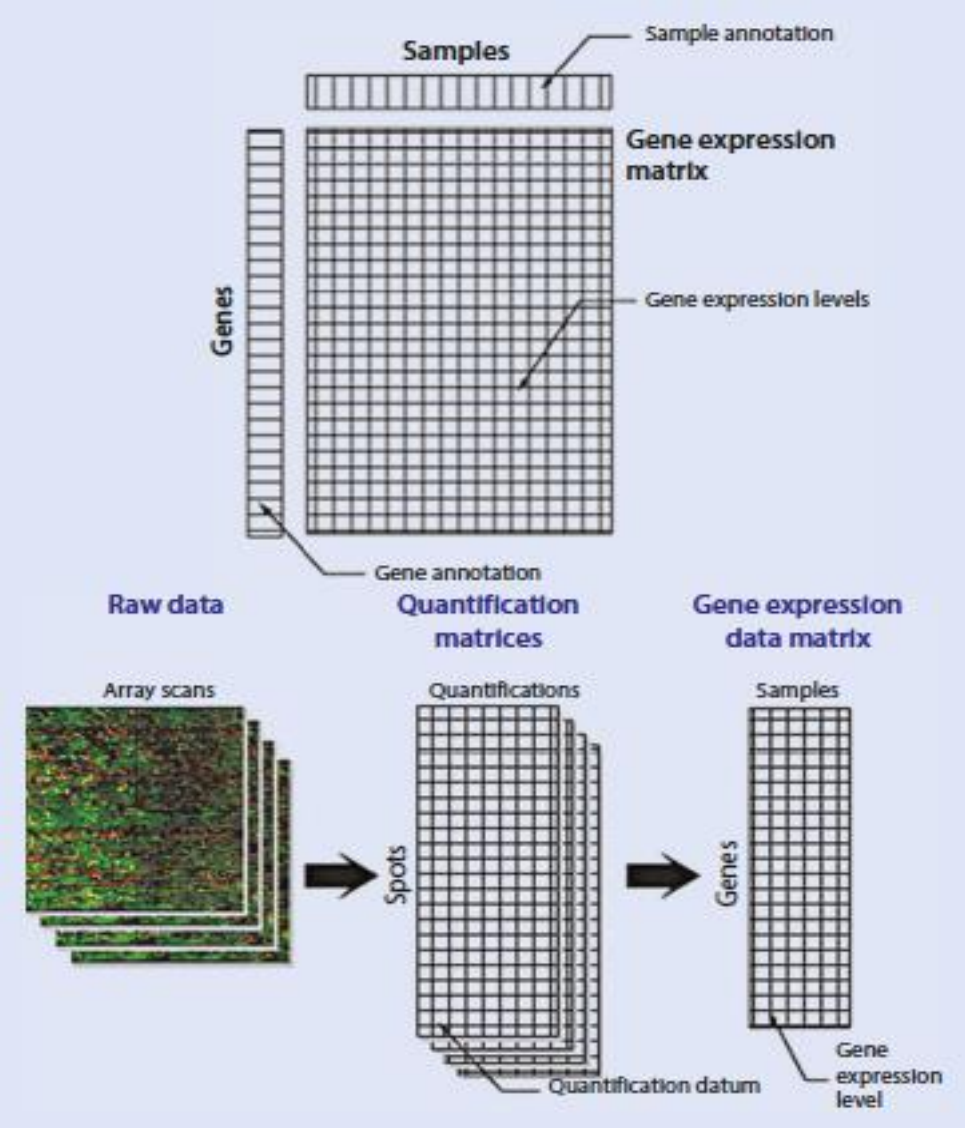


Figure 2.3 Generation and Structure of MA Data

2.7. Machine Learning Approaches

Machine Learning allows machines to learn from experience and improve [45]. According to Lance Brett in his book Machine Learning with R – Third Edition, the abundance of data has led some to call this an era of big data [46]. ML is a field of computer algorithms that can turn data into intelligent action [46]. A rapidly changing environment in which computers, statistical methods, and data availability affect machine learning[46].

We can use ML methods in different engineering and scientific applications to discover knowledge from high-dimensional and nonlinear complex data[47]. We can apply ML methods to gene expression data in identifying cancer[48]. Regardless of the task, machine

learning uses a five-step method to apply the learning process to real-world tasks, and deployment follows these steps[46].

1 ***Data Collection***: This step involves collecting the learning data that an algorithm will use to generate actionable knowledge. Most of the time, it is necessary to combine all of the data into a single source, such as a file, spreadsheet, text, or database[46].

2 ***Data Exploration and Preparation:*** Understanding the data and its nuances are critical[46]. Furthermore, additional work is required to prepare the data for the learning process, which includes cleaning messy data, removing unnecessary data, and recording the data to ensure that it meets the learner's expected inputs[46].

3 ***Model Training:*** After the data preparation for analysis, we will learn what we can learn. The selected machine learning task will inform the selection of the fitting algorithm[46]. We can use this step for supervised ML techniques.

4 ***Model Evaluation:*** In solving the learning problem, the machine learning model results in a partial(i.e., biased) solution; hence, evaluating the model is essential[46]. In evaluating the model, we may use a test dataset or develop performance measures specific to the proposed application[46].

5 ***Model Improvement:*** Utilization of more advanced strategies is necessary to boost the model's performance, and it may be necessary to change to a different type of model totally[46].

After completing these steps, the model can be deployed for its required task if it appears to be performing well[46]. There are three main approaches in machine learning in which we can use algorithms for solving problems, supervised, unsupervised, and semi-supervised. The coming subsections discuss these approaches in detail.

2.7.1. Supervised Learning

To do supervised learning, we need a source of information (X) and a destination (Y). The goal is to learn the mapping between these two things [49]. In this case, we can use supervised learning to train a predictive model for a particular outcome[46]. We can use a labeled dataset for supervised learning. We need to split up the training and test datasets.

Before anything else, the training data is put into the chosen supervised machine learning model to train the chosen model, and this is how we check the model's effectiveness. We look at the test dataset and compare the data's predicted labels to the data's test labels. The supervised learning algorithm makes the model (algorithm) better find the feature values leading to the desired output [46].

2.7.2. Unsupervised Learning

It is different from supervised learning because we do not know the results of unsupervised learning. In unsupervised learning, no one feature is more important than any other. Because there is no one else who can teach the algorithm and there are no labels for the input, the process of making a model is called "unsupervised learning." A system can figure out a function from data that has not been labeled, and this is called "unsupervised learning." The classes are not known yet to learn in this way. Unsupervised learning is hard to evaluate because we do not know the correct answer. Manual evaluation methods are the answer to this problem. Clustering analysis is a way to learn without being told what to do. Unsupervised clustering is used in MA Data analysis to look for genes with similar expression patterns[24].

2.7.3. Semi-Supervised Learning

For classifying data, we can use semi-supervised learning[50]. Semi-supervised approaches use specialized machine learning algorithms to use labeled and unlabeled data efficiently. Whereas unlabeled data is challenging to obtain or cannot be labeled for analysis, semi-supervised machine learning approaches can help. Semi-supervised learning can classify new genes. In pattern recognition and artificial intelligence systems, semi-supervised methods are used extensively[51]. Unlabeled data is used sparingly in semi-supervised learning.

2.8. Application of Machine Learning in Microarray Data Analysis

The availability of public MA Data repositories suitable for applying machine learning approaches in the research is enormously significant. Machine learning methods can enable us to bear on these data, and they can help us discover critical interactions involved in complex experiments. ML methods have cemented their impact on high-level MA Data analysis[47]. These methods have unique performance in dealing with nonlinear interactions

and analyzing large volume and high dimensional data[47]. Using ML techniques, we can get high accuracy and trustful analysis results than the traditional Statistical methods.

2.9. Plasmodium Falciparum Microarray Data

Due to advancements in post-genomic technologies, researchers and scientists can now use microarrays to simultaneously study the expression patterns of hundreds or thousands of genes and proteins. As described in [52], sheared genomic DNA generates the first DNA MA Data for Plasmodium falciparum. They were used to assess differences across the parasite's asexual intraerythrocytic developmental stage.

Oligonucleotide-based microarrays of the P. falciparum parasite were next to completing the P. falciparum strain 3D7 reference genome[53]. These microarrays provided an exceptional in-depth view of the parasite's gene expression changes during the stage of asexual development in the human host's erythrocyte [53]. We can retrieve P. falciparum MA Data from public repositories like GEO from NCBI, PlasmoDB, and ArrayExpress.

2.10. Drug Targets

The majority of medications work by binding to a specific target. Thus, if someone wants to develop a truly novel drug, the first step is to identify potential drug targets. The term "target" is a general term that refers to a variety of biological entities, including proteins, genes, and ribonucleic acids (RNAs)[54]. Additionally, the Oxford Dictionary of Biochemistry, a drug target is defined as "a biological entity (usually a protein or gene) that interacts with, and whose activity is regulated by, a particular compound." This study identifies potential genes that can be used as a drug targets.

2.11. Clustering

Clustering divides a dataset into homogeneous groups or classes[46]. The task determines the clustering algorithm and groups the data without prior knowledge of clustering. Clustering reveals the data's clusters. Clusters share many similarities but differ from one another. It is beneficial to group diverse and high-dimensional data into manageable small groups [46]. We can use Unsupervised learning to classify unlabeled examples.

Clustering tells us which groups of elements are related. We can reduce the data in gene expression analysis to a more understandable or manageable level by grouping the genes into smaller groups and then analyzing those minimized clusters[55]. The primary goal of clustering in MA analysis, is to group genes with similar expression profiles together and genes with different expression profiles together. Gene expression clustering allows for flexible data analysis while not losing individual genes[55]. Within the same cluster, gene expression clusters can infer the functional role of unknown genes.

Co-expressed genes can be grouped with similar cellular functions in MA Data analysis, and these co-expressed genes have the same cellular processes as other similarly expressed genes. Gene co-regulation is more obvious when there is a higher correlation between the expression patterns of the genes in a group. Gene expression data can be clustered by both genes and samples. Gene-based clustering uses samples to represent features, while sample-based clustering uses samples to represent objects.

2.12. Proximity Measurements for Gene Expression Data in Clustering

Proximity (similarity) measurement measures the distance (similarity) between two data objects. The numerical vector, $\vec{V} = \{Gij \mid 1 \leq j \leq NF\}$ represents the gene expression data, where Gij is the value of the j^{th} feature for the i^{th} data object and NF is the number of features in the gene expression data. The distance function of the respective vectors (i.e., $\overrightarrow{Gi}$ and $\overrightarrow{GJ}$) can measure the similarity between two objects (genes) G_i and G_j. This section presents the most commonly used similarity (distance) measuring methods in the clustering of objects.

Cosine Distance: The cosine distance measures the similarity between two vector products if there is a cosine angle between them, and it deals with an orientation of the objects[56]. Given two vectors A and B, the cosine similarity function, $\cos\theta$ function can be defined as in equation 2-1 [56]:

$$\cos\theta = \frac{A.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} Ai*Bi}{\sqrt{\sum_{i=1}^{n}(Aj)^2}\sqrt{\sum_{i=1}^{n}(Bj)^2}} \text{----------Equation 2-1}$$

The cosine distance D between the two vectors A_i and B_j is given by equation 2-2 [56]:

$$D(A,B) = 1 - \frac{A*B}{\|A\|\|B\|} \text{----------Equation 2-2}$$

The time complexity for cosine distance is **O**(3n). It is applicable to document similarity measures[57].

Mahalanobis Distance: The Mahalanobis distance measures the difference between cluster groups of clusters C1 and cluster C2[57]. In this case, the groups can represent samples of specific conditions. Let V be an n-dimensional random vector with the same variance around the mean within the clusters. The difference between their mean vector V can express the dissimilarity between clusters. This type of data measurement is the Mahalanobis squared distance given by the equation below 2-3 [56]. It is applicable for Hyper-ellipsoidal clustering algorithms[57].

$$\boldsymbol{\beta^2 = (U1 - U2)^T \ \Sigma^{-1}(U1 - U2)}$$ **----------Equation 2-3**

In the above equation, β is the metric, T is the matrix transpose, and Σ is the covariance of the given matrix V[56]. From this equation, we can conclude that Mahalanobis distance is related to evaluating covariance matrix inversion, which solves the bias caused by linear correlation[57]. The time complexity for Mahalanobis distance is **O**(3n) [57].

Manhattan Distance: The Manhattan distance measure is a particular case of Minkowski distance measure at u = 1 [56][57]. It is sensitive to outliers. Whenever we use the Manhattan distance measure in clustering algorithms, the shape of the clusters becomes hyper-rectangular. The Manhattan distance D_1, for u, v in n-dimensional space, is given below in equation 2-4[56]. This distance measure is applicable for K-means algorithms[57].

$$\boldsymbol{D_1(u,v) = \|u - v\|_1 = \sum_{z=1}^{n} |u_z - v_z|}$$ **----------Equation 2-4**

where $u = \{u_1, u_2, u_3, \dots, u_n\}$ *and* $v = \{v_1, v_2, v_3, \dots, v_n\}$

The time complexity of the Manhattan distance measure is **O**(n) [57].

Pearson Correlation Coefficient: We can use Pearson correlation in gene expression data clustering. It can measure the similarity between two gene expression patterns[57]. The Pearson correlation measure is sensitive to outliers, which results in false positives by assigning a similar score to a pair of different patterns[57]. The distance measure equation using Pearson correlation coefficient is given by equation 2-5 below, and its application is in partitioning and hierarchical clustering algorithms[57].

$$\mathbf{Pearson(u,v)} = \frac{\sum_{i=1}^{z}(u_i - \mu_u)(v_i - \mu_v)}{\sqrt{\sum_{i=1}^{z}(u_i - v_i)^2}\sqrt{\sum_{i=1}^{z}(u_i - v_i)^2}}$$ ----------**Equation 2-5**

where, $\mu_u \; and \; \mu_v$; are the mean values for u and v, respectively.

The time complexity for the Pearson correlation coefficient measure is **O**(2n) [57].

Euclidean Distance: Euclidean distance is the most commonly used similarity measurement for data clustering[56]. We can use Euclidean distance measurement efficiently when deployed to compact or isolated clusters of datasets[57]. The distance metrics are metrics of Euclidean of squared distance between vector points[56]. Euclidean distance measure's drawback is that, for two data vectors that do not have common attributes, they may result in smaller distances than other pairs of data vectors with common attribute values[57]. The other drawback of the Euclidean distance measure is that the largest-scaled feature could dominate the others. The solution to this problem is applying normalization to continuous features in the data. The squared Euclidean distance between two points; $u_i \; and \; v_j$ for n collection points in z-dimensional Euclidean space that is assigned to the data column matrix $M \in \Re^{dxm}$ is given by[56]:

$M = \{u_1, u_2, u_3, u_4, \ldots u_n\} \; u_i \in \Re^d$

$$\mathbf{z_{ij}} = \left\|\mathbf{u_i} - \mathbf{u_j}\right\|^2$$ ----------**Equation 2-6**

where, $\| \, . \, \|$ Denotes the Euclidean norm[56].

The time complexity for Euclidean distance measure is **O**(n) and, its application is K-means and Fuzzy c-means algorithms[57].

2.13. Clustering Methods

We can apply different clustering methods to MA Data. The main objective of clustering is to find clusters or groups in which instances in one cluster are similar but dissimilar from others. This section presents the most used clustering methods in MA Data analysis.

2.13.1. Hierarchical Clustering

This method is one of the central and old tools used to cluster genes in MA Data analysis[58]. It forms s hierarchical structure in the clusters. This method considers each object as a unique cluster and places the closest objects together into one cluster. Then by merging the nearest branches, it joins all the objects into a single cluster or group. Hierarchical clustering methods can show the final result of clustering as a tree-type graph (also known as

dendrogram), and this can help us interpret the final result easily [58]. The drawback of hierarchical clustering is that if the data have a natural hierarchical feature, it is impossible to trace back to the tree's original branches if there is any mistake in any branch of the structure[58]. There are two types of hierarchical methods:

A. **Agglomerative Hierarchical Clustering:** This method follows a bottom-up approach to cluster the objects (genes in our case). In the beginning, each object represents a cluster of its own, and then the merging of similar clusters of objects stops after achieving the final cluster structure. Figure 2.4 [59] depicts agglomerative hierarchical clustering for N objects. The figure shows that the clustering initially begins with N samples of clusters by considering each object as a cluster of its own. Hence, each cluster contains a single object. Then two clusters with the closest distance will merge until the number of groups becomes one or as specified by the user[59]. From the figure, we can conclude that the leaf nodes are individual elements of the dataset, and the root represents the entire dataset.

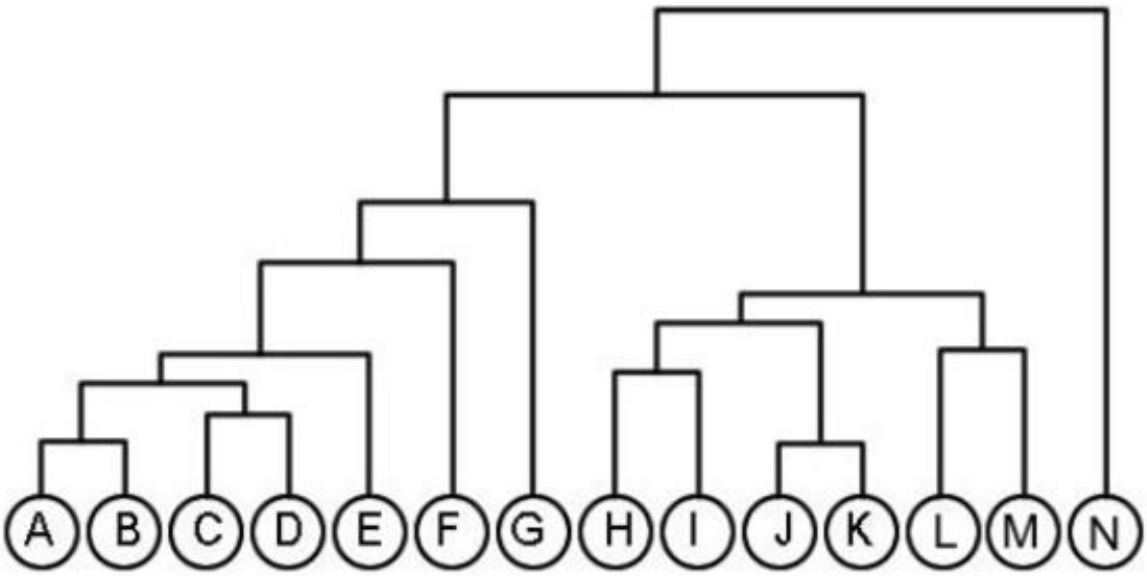


Figure 2.4 Dendrogram Representation for an Agglomerative HC

B. **Divisive Hierarchical Clustering**: It is a top-down approach in which all the objects at the beginning belong to a single cluster[59]. It starts with an all-inclusive cluster that contains all the objects for clustering. Iterative partitioning of existing clusters can form sub-clusters, and one cluster is chosen at a time and split into two subclusters. Divisive hierarchical clustering is good at identifying large clusters.

2.14. Gene-Gene Interaction Network

The gene-gene interaction network gives information about the kind of interaction within the genes found in the DNA of living organisms. The detection of gene-gene interaction networks from gene expression data becomes an exciting area of research[60][61][62]. The expression of some genes from the parent in the child leads to specific physical characteristics, and other genes may remain silent, leading to the absence of specific physical characteristics. This interaction within the genes can form a virtual network known as a gene regulatory network or gene-gene interaction network[60]. The gene-gene interaction network is significant in many biological diseases[60]. It helps to find the controlling genes responsible for diseases like cancer, which helps to stop the disease.

The identification of gene-gene interaction networks is a data-driven problem. The gene expression data helps us determine the reason for its over or under expression across the samples[60]. The expression of the gene can encode the regulator. The functional product of some other gene can regulate the gene expression; this results in some complex gene-gene interaction network that can form the structure of the entire gene-gene interaction network[62]. The task of constructing a gene-gene interaction network is challenging because living organism cells have thousands of genes with which each can interact with the regulators encoded by one or more other genes[62]. Researchers face two challenges while constructing a gene-gene interaction network[62]. The first challenge is the presence of noise in the gene expression data and the second one is the bareness of samples compared to the high number of genes. Different researchers propose machine learning methods to construct gene-gene interaction networks [60]–[67].

2.15. Related Works

This section introduces a comprehensive review of essential related works and the most recently published scientific research focusing on MA Data analysis that uses machine learning methods. Here we can clearly understand the general technique, methods, and results of existing studies around MA Data analysis.

Using stage-specific metabolic network analysis, Neel Dholakia, Pinakin Dhandhukia, *et al*. [25] identified potential target of Plasmodium falciparum using MA Data. In GEO, they

found GSE24416, a gene expression dataset. Then they looked at the MA Data to find differentially expressed genes. Then they used the R Bioconductor Statistical package limma to normalize and transform the data. They used CLC Main Workbench to calculate ANOVA for differential gene expression. They excluded genes with a p-value of 0.05 or higher and fold change > 1. Their results were plotted on the P. falciparum pathway using Omics Viewer software after filtering out genes that were not significant. They used Boxplot to compare probe intensity variation between arrays. From 5100 genes, 3021 are DEG. They made a volcano plot with the Y-axis representing the significance of differentially expressed genes and the X-axis representing fold-change. They overlaid differential expression results with metabolic pathways to find pathways required for parasite progression through its erythrocytic (red blood cell) stages. In order to reconstruct the metabolic network, they used flux balance analysis. Finally, their research identified a putative targe as the crucial component for the parasite's stage transition. The gap in this research is, under tight assumptions about the nature of the microarray data, it is impossible to do an ANOVA analysis properly. The fact that there is no unique interpretation of the significance of two means of the genes makes it less useful than the t-test, which is a more powerful test. Further testing is necessitated by the requirement of the post-ANOVA t test.

A. Bustamam, S. Formalidin, et al. [68] used Lymphoma microarray data for testing the best clustering using Davis Bouldin Index. Their goal is to cluster and analyze the characteristic mode of matrix C using SVD from lymphoma gene expression data and to display the best cluster in a dendrogram based on AHC. They also compared their clustering result with another clustering method called Co-Similarity. They propose using Singular Value Decomposition (SVD) to cluster gene expression data from Lymphoma, denoted by matrix X, into two matrices, R and C, each of which has a size of 4026 x 62 and 62 x 62. Then they find the cluster and determine the number of members in each subcluster using Agglomerative Hierarchical Clustering (AHC) and displaying the best cluster in a dendrogram based on hierarchical clustering. Their results show that reducing the dimension of the characteristics mode matrix C in the decomposition matrix X of gene expression data to three gets the best cluster according to DBI value. A three-dimensional matrices characteristic mode C had the same number of members in each of its subclusters. Using SVD to lower the rank of matrix X gives better clusters than using the Co-Similarity method.

Massoume Eskandari, Shahla Chaichian et al. [26] analyzed ovarian cancer microarray data to find a pattern for detecting gene expression changes in ovarian cancer using microarray technology. After normalizing microarray data using four different normalization methods (LOESS, 3D LOESS, NN3, and NN4), they evaluated three clustering algorithms (k-means, fuzzy c-means, and hierarchical) to determine the best, most efficient algorithm. They use a robust microarray analysis algorithm to process data files containing probe-level intensities for background adjustment, normalization, and gene identification. They normalize the data using locally weighted least squares regression (LOESS). Their findings indicated that using K-means and fuzzy C-means for gene expression is not optimal. As a result, they consider hierarchical clustering, which identified three genes.

Micheal Olaolu Arowolo1, Marion O et al. [69] analyzed the gene expression of a malaria vector. Their study used the PCA feature extraction algorithm to extract latent components from a high-dimensional malaria vector RNA-seq dataset and the Ensemble classifier to evaluate the resulting dataset's classification performance. They tested the experiment's effectiveness using an RNA-Seq dataset from the mosquito Anopheles gambiae. They obtained the data from western Kenya and consist of mosquito genes from 2010 to 2012. They used 2457 instances with seven attributes of genes. This study uses PCA on Anopheles mosquito data, revealing valuable gene information for further research. They implemented the model using MATLAB's Ensemble Adaboost algorithm. In 7.8486 seconds, PCA extracted 1592 significant gene features and 45 latent components.

Priyanka Ramesh, Shanthi Veerappapillai et al. [70] analyzed Severe Acute Respiratory Syndrome coronavirus Microarray Data. Differential gene expression, protein-protein interaction, and Gene Ontology (GO) studies were done on SARS-CoV datasets GSE1739 and GSE33267 to understand better how the host's immune system responds to the pathogenic coronavirus and thus cut down on death rates. They used P-values and log2 fold change values less than 0.05 and 1.5 to look for DEGs in their data set. They found 79 of them. Network and GO analysis found that two hub genes, ELANE and LTF, had different levels of expression in the lungs, leading to more pro-inflammatory cytokines in the lungs. They knew that ELANE and LTF are not the exact causes of a cytokine storm, which leads to death. Targeting the genes that cause cytokine storms may be an excellent way to fight SARS-CoV-2 infection and reduce deaths. Predicting how COVID-19 will do in the future could help design an immune intervention soon, cutting down on deaths.

2.15.1. Summary of Related Works

The following Table 2 summarizes some of the related works to the general concepts of MA Data analysis. This study addresses the gap mentioned below in the table. In [68] they did not perform data transformation. Differential gene expression analysis expects log transformed data. The data transformation can stabilize the variation across the range of signal intensities. It reduces the complexity of the data matrix and makes it better organized. This research addresses the gap with not transforming the data, by transforming it into log base 3 form. In [26] they did not validate their clustering result in which their study misses the vital issues essential to the success of the clustering application. The validation can measure the goodness of the clustering result. This research can address the gap mentioned in by [26] validating the clustering result using internal validation technique. In [69] they did not perform feature selection. Feature selection microarray data analysis is considered as selecting the most significant genes. In addition to this they used PCA that can produce overlapped clusters. Our study addresses the gap bey selecting the most informative genes using differential expression analysis. The study can also solve the problem of having overlapped clusters produced by producing tight clusters using hierarchical clustering technique.

Table 2-1: Summary of related works

Ref No.	Title (Author/s)	Year	Dataset	Methodology	Gap
[68]	Clustering and Analyzing Microarray Data of Lymphoma Using Singular Value Decomposition (SVD) and Hybrid Clustering (A. Bustamam, S. Formalidin, et al.)	2018	Lymphoma Microarray Data	Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF)	→ They did not perform data transformation
[26]	Identifying the Most Appropriate Pattern for Identification of Gene Expression Changes in Ovarian Cancer Using Microarray	2019	Ovarian Cancer Microarray Data	k-means, fuzzy c-means, and hierarchical	→ They do not validate their clustering models
[69]	An Efficient PCA Ensemble Learning Approach for Prediction of RNA-Seq Malaria Vector Gene Expression Data Classification	2020	Mosquito Anopheles Gambiae	PCA, MATLAB's Ensemble Adaboost algorithm	→ They did not perform feature selection
[70]	Gene expression profiling of coronavirus microarray datasets to identify crucial targets in COVID-19 patients (Priyanka Ramesh, Shanthi Veerappapillai, et al.)	2021	Severe Acute respiratory syndrome coronavirus Microarray Data	They used P-values and log2 fold change for DGE.	→ They did not perform any validation on their result

CHAPTER THREE

3. The Research Methodology

This chapter discusses the research methodology and the steps involved in analyzing P. falciparum MA Data using machine learning approaches. It discusses the systematic literature review approach we used in this study and the P. falciparum MA Data retrieval and description. Then it discusses the study's MA Data analysis methods and tools. The final section of this chapter clarifies the machine learning model for MA Data and its evaluation technique. This study follows an experimental research method, in which we performed experimentation for finding differentially expressed genes, experimentation for clustering genes with the same biological behavior and experimentation to find potential drug targets.

3.1. Systematic Literature Review (SLR)

Systematic Literature Review is a technique to identify, evaluate, and summarize a particular topic's advancement in the literature. It enables the collection of specific literature, allowing for detailed methodological analysis with less bias than traditional reviews. The goal of SLR is to develop a broad-spectrum vision of a specific question and provide a good summary of the literature. We adopted the approach we followed for the SLR from Diego C. B. Mariano et al.[71] and B. Kitchenham[72] guidelines. In this study, the literature review has gone through the following steps, as shown below:

- → In the first step, we write down vital information, like the research questions and goals, to find and collect relevant publications.
- → Then we establish a list of keywords to collect our references. References are collected from the literature of available publication repositories while emphasizing publications from top sites.
- → We used exclusion criteria to the search results to remove irrelevant papers.
- → We grouped the sets of results from each search result.
- → Then we evaluated our collected references and applied simple filtration on the chosen papers.

3.1.1. Search Strategies

We applied a generic search for our study using the following aspects:

1. Search for synonyms and other related words.

2. Boolean “OR” used to relate synonyms
3. We used the Boolean “AND” for the grouping of essential terms.

Table 3-1 Search Strategy Tables

Table 3-1 A Search Menu Extracted From Our Study's Main Idea

Concept 1	Microarray Data
Concept 2	Analysis
Concept 3	Machine Learning Approaches
Concept 4	Gene-Gene Interaction

Table 3-1 B Synonyms in the Search Menu

Concept 1	Microarray Data Gene Expression Data
Concept 2	Analysis
Concept 3	Machine Learning

Table 3-1 C Search Table for Applying Boolean AND

Concept 1	Microarray Data OR Gene Expression Data OR Transcriptome Data OR Genome Data
Concept 2	Analysis
Concept 3	Machine Learning
Concept 4	Gene-Gene Interaction

3.1.2. Search Strings

We perform the top search string using different keywords within logical connectives. The following table depicts the search string strategy.

Table 3-2 Search String Items

Search Strings
("All Metadata”: Microarray Data) OR ("All Metadata”: Gene Expression data)
AND ("All Metadata”: Analysis)
AND ("All Metadata”: Machine Learning) OR ("All Metadata”: Clustering) OR ("All Metadata”: Classification)
AND ("All Metadata”: Gene-gene interaction network) OR ("All Metadata”: Gene regulatory network)

We search publications from the following platforms:

→ NCBI
→ IEEE Xplore Digital Library
→ Springer Link
→ ACM Digital Library
→ Google Scholar
→ ArrayExpress
→ Nature
→ Research Gate
→ Hindawi

We conducted the last search on 26th November 2021, and we used publications published up to this date.

3.1.3. Inclusion and Exclusion Criterion

We filtered out papers using the inclusion and exclusion criteria. We removed papers that did not meet the exclusion criterion and kept papers that met all the inclusion criteria for further investigation. In this study's SLR, we use the following Inclusion and Exclusion criteria:

→ We excluded non-English papers.
→ We identified duplicate papers and then excluded the duplicate ones.
→ We finally excluded papers that are not related to MA Data.

3.2. Microarray Data Repository

A MA Data repository is a database that holds gene expression data from microarrays. The repositories can store the data, enable users to search for the data using a searchable index, and make the data available for analysis and interpretation according to the user's interest. We can access the data directly from the repository or download it for interpretation. There are two types of MA Data repositories that allow users access:

1 Public or peer-reviewed MA Data repositories: these repositories can provide free access to datasets. These repositories adhere to public or industry standards. ArrayExpress from EBI, PlasmoDB, and GEO from NCBI are examples of this data repository.

2 Specialized MA Data repositories: these repositories function for a specific purpose, and they can be commercial, academic, or non-profit. These repositories have their brand for a particular entity, a method of data analysis, or a set of functions.

With a large amount of data produced by microarray studies, storing the dataset in public repositories is essential to enable researchers to use the data to complement and enhance their studies.

3.2.1. Plasmodium Falciparum Microarray Data Collection

We download the dataset for this study from GEO of NCBI with GEO accession number GSE18780[73]. It is a file generated by Affymetrix GeneChip Scanner 7GPlus image analysis software. The experiment type of dataset is expression profiling by microarray. The dataset contains the expression profile of P. falciparum Anopheles gambiae (African malaria mosquito) gene parasites from the midgut and salivary glands; gene expression profile of mosquitoes infected vs. uninfected midguts and salivary glands. It contains a total of 12 samples and 22679 genes(variables). Three samples are from the mosquito's midgut and are noninfected (i.e., control samples). Three samples are from the mosquito's midgut, infected with P. falciparum. Three samples are from the mosquito's salivary gland, noninfected samples. Three samples are from the mosquito's salivary gland, infected with P. falciparum. Further detail about the dataset found at the NCBI website with the accession number GSE18780.

3.2.2. Microarray Data Structure

Understanding the nature and properties of the data structure produced by microarrays is critical. The organization of MA Data is as a matrix, with rows representing genes and columns representing samples or experimental conditions. Each data point in the matrix represents the level of expression of a specific gene in a given sample or experimental condition, and a row or column of entries can form an expression pattern.

3.3. Microarray Data Analysis Methods and Tools

This section discusses the general methods for MA Data analysis and the critical tools required to complete this study. The following figure 3.1 illustrates the general workflow for analyzing MA Data used in this study.

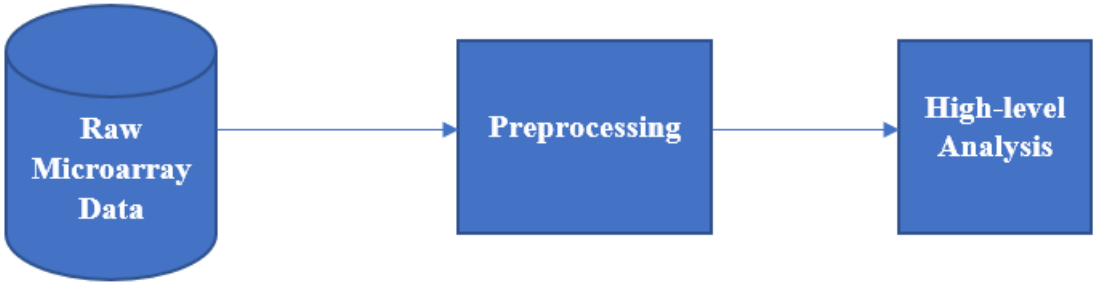


Figure 3.1 General Workflow of MA Data Analysis

The above Figure 3.1 depicts that the starting point for the MA Data analysis is the raw MA Data. After downloading the raw MA Data, the preprocessing step comes into the line. We can conduct high-level analysis on the preprocessed data. We can apply different analysis algorithms in high-level analysis, including basic statistical methods and advanced machine learning algorithms (i.e., unsupervised). The last thing in the high-level analysis is constructing a gene-gene interaction network.

3.4. Development Tools

This section depicts the necessary tools and packages to realize the proposed study's functionality.

3.4.1. Design Tools

To draw diagrams in this research paper, we used Drawio open-source software. This software helps users to create and publish different kinds of diagrams reliably. It provides different features to draw workflows, flow charts, UML diagrams, business diagrams, database diagrams, and ERD[74].

3.4.2. Packages

The following table shows some of the packages used and their purpose in this study for the successful implementation of the microarray dataset.

Table 3-3 Packages Used in this Study

Package	Purpose
limma	Used for differential expression analysis

genefilter	Provides methods for filtering genes
RColorBrewer	Used to create the primary visual differences between classes
arrayQualityMetrics	For quality assessment of the microarray
DataExplorer	For exploratory analysis the data
cluster	Method for cluster analysis
clValid	Used for validation measure for clustering results

3.5. Data Preprocessing

Microarray gene expression data is a measured intensity that requires background correction and normalization before any kind of data analysis[75]. The starting point for preprocessing this study's P. falciparum MA Data is the CEL file that stores the intensity value obtained from the GEO with the accession number GSE18780[73]. Microarray experiment results from the Affymetrix GeneChip Scanner 7GPlus software are in the CEL file. It contains all of the data from the experiment, which they got from the software. Intensity is the numerical value of a pixel representing genes' expression values of the parasite's midgut and salivary gland.

Raw MA Data is not always the best choice for gene expression analysis. The various stages, such as labeling, hybridization, and scanning, can cause noise and technical variability in the data[76]. We can apply preprocessing methods to the raw MA Data for reducing the background noise, for normalizing the data, for controlling the quality of the data, for filling missing values, outlier detection, and for transforming it into an acceptable format and feature selection for the selected analysis method[47] [75] [76].

We can consider all the preprocessing steps as low-level analysis stages. We used the R package of limma to preprocess our MA Data, including background correction, normalization, and final probe summarization. The preprocessing step aims to get precise and accurate gene expression data for high-level analysis.

3.6. High-level Analysis

Once we preprocessed our raw data, high-level analysis methods can come into the line to explore biologically important information from the data. The high-level analysis objective is to find DEGs across samples, cluster genes in which they are close to each other, and

derive a gene-gene interaction connectivity model that fits the P. falciparum microarray dataset. The hypothesis at this level of analysis is that the expression levels of genes at a given point are a function of the expression levels of genes at the previous point sample. The methodology for the high-level analysis is adopted from the study by Yongliang Yang, S. James Adelstein *et al*. [24].

3.7. Machine Learning Approach

In high-level MA Data analysis, we can employ the machine learning approach. Machine learning has recently gained popularity for analyzing high-dimensional MA Data. [77]. This study can use an unsupervised machine learning technique, hierarchical algorithms.

3.8. Machine Learning Model for Microarray Data

This section discusses the steps involved in applying the machine learning model to MA Data. There are two steps involved in developing a machine learning model for MA Data analysis [78]. The following subsection discusses the steps in detail. The assumption made before developing the model is that the data is preprocessed and ready for high-level analysis.

3.8.1. Learning from Data

The predominant feature of machine learning is that learning from data learning is simply the process of determining a machine learning model's decision variable. There are three types of machine learning paradigms in analyzing MA Data. The first is unsupervised machine learning, supervised machine learning is the second, and the third is the semi-supervised machine learning paradigm. We use the unsupervised machine learning paradigm in this study.

3.8.2. Clustering Algorithms Evaluation

There is no biological criterion for selecting an algorithm, and we can evaluate our clustering results using orthogonal information. If functionally related genes fall in a similar cluster, we can judge our clustering depending on its biological predictive value (this evaluation criterion is beyond this research). So, in this study paper, we used a mathematical method to evaluate our clustering machine learning algorithm.

CHAPTER FOUR

4. The Proposed Solution for P. falciparum Microarray Data Analysis

4.1. Overview

This chapter depicts preprocessing and high-level analysis of P. falciparum MA Data in-depth to show each component discussed in the proposed methodology to solve the problem understudy.

4.2. The Proposed Plasmodium Falciparum Microarray Data Analysis Architecture Framework

The conceptual framework of the proposed MA Data analysis shows the steps we follow to solve the problem. We depict the conceptual framework diagram for preprocessing step and high-level analysis separately. Figure 4.1 shows the conceptual framework of the proposed preprocessing step of the MA data analysis.

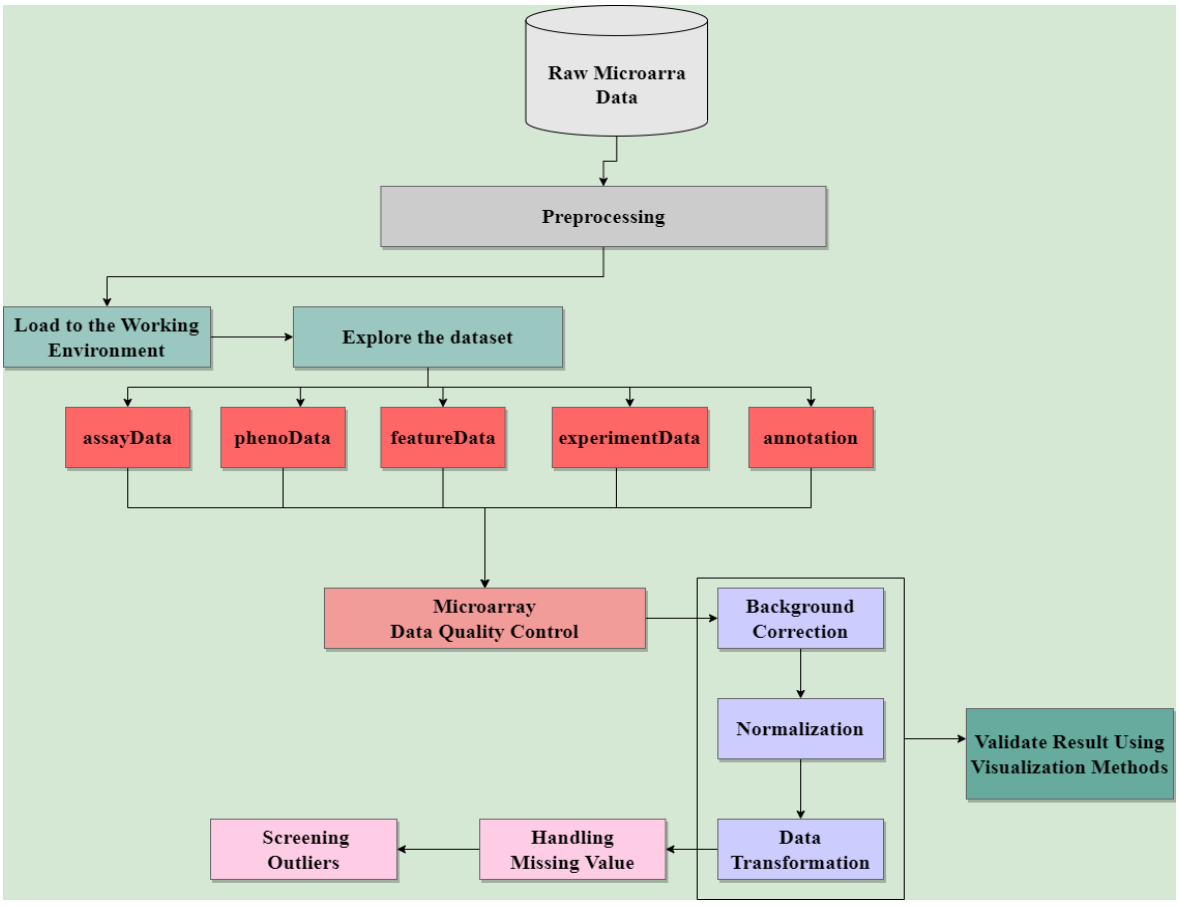


Figure 4.1 Conceptual framework for preprocessing raw MA Data

The above diagram shows that the first step is to search for the raw MA dataset from MA data repositories. Then the next step in the preprocessing is to load the raw MA Dataset into the working environment of the R-studio. We then look through the dataset to see if there are assyData, phenoData, featureData, experimentData, and the annotation file. Section 5.4.2 presents further information about those data.

Then we check the quality of the raw MA data. This step will generate the array quality report, including information about array comparison, the array intensity distribution, variance mean dependence, Affymetrix specific plots, and information about individual array quality. After getting the above information, we perform background correction, normalization, and transformation of the data. The other thing to do in this step is to visualize the result for validation and further improvement if required. Pre-processing includes missing value imputation and outlier screening. Then the data becomes ready for high-level analysis.

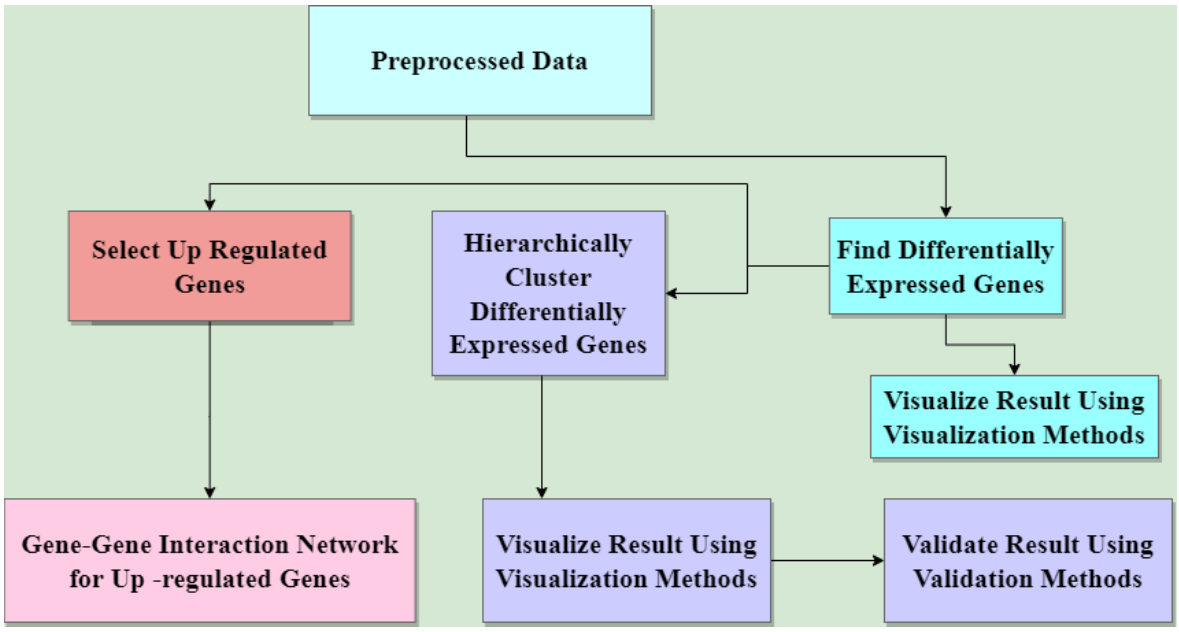


Figure 4.2 Conceptual framework of the proposed high-level MA Data analysis

The above Figure 4.2 depicts the conceptual framework of the proposed high-level MA Data analysis. The high-level analysis uses the preprocessed data as an input. The first step in the high-level analysis is to identify differentially expressed genes. We visualize the results of the differentially expressed genes using different plots. Then we cluster those differentially

expressed genes to group genes with the same biological behavior. This research uses hierarchical clustering techniques. Then we visualize the result of clustering. The last step in clustering is to validate our model and select the best model. The last stage of the high-level analysis is to drive the gene-gene interaction network for the up-regulated genes. This interaction network can reveal the biological information of particular genes, and we can extract a list of potential drug targets from the interaction network.

4.2.1. Preprocessing of the Raw P. falciparum Microarray Data

This section describes an overview of the preprocessing of gene expression data generated using microarray technology. The preprocessing step helps to get biologically meaningful, accurate, and precise gene expression data for further analysis. Generally, the preprocessing step involves importing the raw intensity data, background correction, normalization, quality control, and transformation. The following sub-sections give further explanation.

4.2.1.1. Reading or Importing Data

We can import the MA Data to the working environment of R studio using two methods. The first method is reading or importing an output file that results from an image analysis software. The second method is importing a data frame of expression values directly from a file or data as an R object. In this study, we import the dataset into the R studio using the first method in which the data is the result of an image analysis software Affymetrix GeneChip Scanner 7GPlus. We can import the raw image files using the limma package from the Bioconductor.

4.2.1.2. Microarray Data Quality Control

Quality control of MA data is an essential step for the downstream analysis, and it helps us check the existence of anomalies in the data. We can visually inspect the MA data using diagnostic plots to spot possible quality problems, including degraded samples or array handling or sample processing problems. The diagnostic plots can show biases that we should remove before further analysis. Algorithm 4.1 illustrates the steps involved in determining the quality of MA data.

```
start
    load libraries
        load Raw GES18780 Data
        Use Array Quality Metrics Method
        Browes the Result using Web Browser Software

end
```

Algorithm 4.1 Algorithm to Determine the Quality of the GSE18780 MA Data

4.2.1.3. Background Correction

When importing an array image, it is common to read both background and foreground intensities for each probe. We can estimate the ambient intensities that can affect each probe using the background intensity values. In the beginning, background correction removes a non-specific signal from the background intensity of each probe. The limma packages have a lot of complex ways to remove the background from raw microarray images. Algorithm 4.2 shows the steps for applying background correction.

```
start
    load libraries
        load Raw GES18780 Data
        Applay Background Correction Method

end
```

Algorithm 4.2 Algorithm for Applying Background Correction

4.2.1.4. Normalization

Normalization ensures that all samples are measured on the same scale, simplifying downstream analysis. Normalization aims to eliminate systematic effects caused by technical differences between microarray assays unrelated to the biological changes of interest. The Bioconductor package limma includes numerous normalization methods for MA Data. This study proposes quantile normalization to remove unwanted technical variation and reduce bias in detecting true differences between samples. Algorithm 4.3 illustrates the steps required to normalize the data.

```
start
    load libraries
        load Raw GES18780 Data
        Applay Normalization Method
        check if the Data is Normalized
            if(GSE18780 Data is Normalized)
             end
            else
            Go to Line Number 4
end
```

Algorithm 4.3 Algorithm for Normalization of GSE18780 MA Data

4.2.1.5. Data Transformation

Data transformation is the logarithmic transformation of the MA data. The main objective of data transformation is to stabilize the variation across the range of signal intensities. Data intensities with maximum values in microarrays can cause more significant variation. Data transformation can help to reduce the complexity of a data matrix. Transformation to normality, centralization, and re-scaling are three critical components of MA Data transformation. The term "transformation to normality" refers to the process of adjusting MA Data to approximate a normal distribution. Centralization removes data bias from the data. Re-scaling the data ensures that the data variances from different hybridizations are equal. Algorithm 4.4 shows the steps for applying data transformation.

```
start
    load libraries
        load Raw GES18780 Data
        Applay log2 Data Transformation Method
        check if the Data is Transformed
            if(GSE18780 Data is Transformed)
             end
            else
            Go to Line Number 4
end
```

Algorithm 4.4 Algorithm for GSE18780 Data Transformation

4.2.1.6. Feature Selection

Remove features that are no longer relevant or redundant with the aid of feature selection. Finding a set of DEG across the samples is the same as selecting features in the MA Data analysis[79]. This research paper performs the DEA in the high-level analysis step.

4.2.2. High-level Analysis

This section discusses the proposed high-level analysis of the P. falciparum MA data. The following subsection discusses the details for identifying DEGs, clustering the genes, and lastly, the gene-gene interaction network.

4.2.3. Differential Expressed Genes Identification

This study uses the limma library from the Bioconductor package and the steps in GEO2R from the NCBI official website to identify DEGs across samples. The identification of DEGs aids in the discovery of the most significant genes across all samples. This study proposes identifying 2500 DEGs out of 22769 genes based on P-value and using the false discovery rate arrangement. Algorithm 4.5 shows the steps for identifying the DEGs across samples.

```
start
    load libraries
        Use Preprocessed GES18780 Data
          Make Proper Column Name
             Assign Group Membership of the Samples
                 Split the Group Membership
                     Assign Samples to Groups
                          Design the Matrix for Linear Model
                              Adjust Contrasts of Interest
                                  Recalculate Model Coefficients
                                      Compute the Statistics for DEG using Emperical Bayes function
                              Extract the Top Ranked Genes
                         Clean the Data to Remove Missing Values
                     Check if the Data is Clean
                   if(Data is Clean)
                 Visualize the Result
        else
    Go to Line 13
end
```

Algorithm 4.5 Algorithm to Identify DEGs

4.2.4. Clustering

This study proposes data clustering algorithms. It suggests hierarchical. The clustering results show that genes within the same groups behave biologically the same. This step's top significant genes are the input data for clustering, and the study also suggests internal validation of the clustering model. Algorithm 4.6 depicts the steps for gene clustering.

```
start
    load libraries
        Use Preprocessed DEG Data
          Visualize the Data
             Randomly Select Two set of genes with Three Elements in Each Set
              (Clustering Method = Hierarchical)
                Calculate Distance Between Genes
                    Cluster the Genes based on their Closeness
                        Plot the Clustering Result
                            Cut the Dendrogram at Some Point
                                Retrive Cluster Indices of Genes
                                    Visualize the Clustering Result
          Validate Clustering Techniques
end
```

Algorithm 4.6 Algorithm for Clustering DEGs

4.2.5. Gene-gene Interaction Network

This study proposes implementing a gene-gene interaction network using the free and open-source Cytoscape software available online. Cytoscape can visualize interaction networks and biological pathways, and it integrates the network results with other information such as annotation files, gene expression profiles, and other sources of information. This study proposed the construction of the gene-gene interaction network using some of the software's default values. The input data consists of a list of upregulated genes, each with an associated expression level.

CHAPTER FIVE

5. Implementation and Experimentation

5.1. Overview

This chapter describes the detailed implementation of the P. falciparum MA data analysis using the methodology discussed in the third chapter. It also implements the proposed solution discussed in the previous chapter. This study follows an experimental approach to analyze the MA Data using machine learning algorithms. Precisely in this chapter discusses both the preprocessing step and high-level analysis implementations in the best possible way. First, for performing the preprocessing step, we use the limma Bioconductor package, and then for the high-level analysis, this study uses statistical methods, machine learning algorithms, and Cytoscape software.

5.2. Implementation Environment

In order to implement the proposed solution, this research uses some development tools and R packages.

5.3. Deployment Environment

We deployed the tools stated in the implementation environment on a personal computer equipped with ACER NITRO 5 Intel(R) Core (TM) i5-9300H CPU @ 2.40GHz, 2400 MHz, 4 Core(s), 8 Logical Processor(s). The operating system of the computer is Microsoft Windows 11 Home. Installed Physical Memory (RAM) 8.00 GB.

5.4. Implementation of Preprocessing Step

The study uses the R programming language to implement P. falciparum MA data preprocessing. To accomplish the study, we perform the following activities: installing R Packages from Bioconductor and CRAN, loading different libraries to the working Environment of R studio, loading the dataset to the disk, and using the methods in the libraries we perform the preprocessing. The dataset loaded from the NCBI is used throughout the whole implementation of this study.

5.4.1. Loading the Dataset to the Environment

We began by searching for the MA dataset on the NCBI website, which had the GEO accession number GSE18780 and then loading the dataset into our working environment, R studio. We placed the dataset in a folder that we had created specifically to store the R document. The following sample code shows a complete representation of the loading of the MA data GSE18780.

```
1  # Load The Microarray Data to The Working Environment
2  filelist <- read.delim("filelist.txt",check.names=FALSE,as.is=TRUE)
3  rownames(filelist) <- filelist$Name
4  PFcelFiles <- affy::list.celfiles()
5  PFcelFiles
6
7  PFcelFiles %in% filelist[,2]
8
9  GSE18780 <- ReadAffy(phenoData=filelist, verbose = TRUE)
10
11 sampleNames(GSE18780)
12 print(GSE18780)
13 class(GSE18780)
14
```

Figure 5.1Sample Code for Loading and Checking GSE18780 MA Data

In the above code, the variable GSE18780 stores the MA Data. Line 2 loads the text file, which contains the sample names as a text file. Line 7 checks whether the filenames are the same in the directory and the sample info tab of the text file. Line 9 reads the raw MA Data and stores it in the GSE18780 variable. Line 12 prints the details of the AffyBatch object GSE18780. Line 13 prints the class of the data.

5.4.2. Exploring the Dataset

When analyzing a high-dimensional dataset for the first time, it is essential to explore it to see what structures or patterns are present. The existence of various bits and pieces makes MA Data very complex. Bioconductor can provide a way to manage the MA data by storing the dataset in a single structure. The Biobase package contains methods or functions that can help to represent the standardized MA data structure. Different sources of information about the MA data can be combined using the ExpressionSet class from the Biobase package. We manipulated the ExpressionSet and used it as an input or output to various Bioconductor functions used in this study. The data in ExpressionSet class consists of the following data:

→ ***assayData***: Expression data derived from microarray experiments are stored here. A matrix of expression values or an environment can represent assayData. Rows can represent features (probe sets) and columns samples in an assayData matrix. Using the exprs () function, we can get the assayData passed to it. A component's name should be "exprs" in the assayData if it is an environment with identically sized matrices. As a matrix, the exprs () function uses to get at the "exprs" element. This table must have distinct rows and columns with names corresponding to those in featureData and phenoData. The assayData can be accessed using the following code.

```
GSE18780@assayData
GSE18780@assayData[["exprs"]]
```

Figure 5.2 Sample Code for Exploring the assayData

→ ***phenoData***: This information pertains to each sample in the experiment, and it is a data frame with optional annotations. The row count in phenoData must match the column count in assayData, and phenoData's row names should match the matrix's column names in the assyData. The following sample code can explore the phenoData.

```
GSE18780@phenoData
GSE18780@phenoData@varMetadata
GSE18780@phenoData@data
GSE18780@phenoData@.__classVersion__
```

Figure 5.3 Sample Code to Explore the phenoData

→ ***featureData***: Additionally, this is an optional annotated data frame that contains information about the dataset's characteristics. The row numbers and names in featureData should correspond to the row numbers and names in the assayData matrices. The following sample code explores featureData.

```
GSE18780@featureData
GSE18780@featureData@varMetadata
GSE18780@featureData@data
GSE18780@featureData@.__classVersion__
```

Figure 5.4 Sample Code to Explore featureData

→ ***experimentData***: It is optional and describes the experiments in detail. The following sample code can explore the experimentData.

```
GSE18780@experimentData
GSE18780@experimentData@other
GSE18780@experimentData@.__classVersion__

```

Figure 5.5 Sample Code to Explore experimentData

→ ***annotation***: It is the name of a Bioconductor chip annotation package that explains the platform information used to analyze samples. We can check the annotation of the microarray chip using the following sample code.

```
GSE18780@annotation

```

Figure 5.6 Sample Code to Check annotation

5.4.3. Microarray Data Quality Control

Checking the data quality from the array is an essential part of data analysis. In this research, we performed the quality assessment before preprocessing the data to get a clue in following the most appropriate preprocessing steps that can be essential to correct the most dominant effects found in the raw data. The following sample code can generate a microarray quality assessment report of the raw GSE18780 MA Data.

```
arrayQualityMetrics(GSE18780,
                    outdir="GSE18780_Quality_Assesment",
                    force = T)
browseURL(file.path("GSE18780_Quality_Assesment", "index.html"))

```

Figure 5.7 Sample Code to Check the Quality of GSE18780 Microarray

5.4.4. Background Correction, Normalization, and Data Transformation

Background correction compensates for the ambient intensities surrounding each feature in the dataset by adjusting the MA data. It can adjust the neighborhood of each microarray feature, whereas normalization helps to get reliable and usable data from microarrays; we must normalize the raw data prior to performing downstream analysis. We can adjust the MA data for effects that can arise from differences due to technology between the printed

probes using the normalization technique. Normalization can help us to make measurements from different arrays comparable. We used the *rma()* function to perform both background correction and normalization of the data. Besides performing background correction and normalization of the raw data, the *rma()* function can calculate the expression values. The function can also transform the expression measure into a log with a base two scale. The *rma()* function can store the final result as an ExpressionSet object. The following sample code depicts this step.

```
71  NorBgckGSE18780 <- rma(GSE18780, subset=NULL, verbose=TRUE,
72                         destructive=TRUE, background=TRUE,
73                         normalize=TRUE, bgversion=2)
74
```

Figure 5.8Background Correction, Normalization, and Data Transformation

5.4.5. Handling Missing Value

Missing value imputation is a critical step in MA Data analysis, and missing data points can impact downstream gene expression analysis. First, we used the function as.matrix() to convert the background-corrected, normalized, and transformed data to a gene expression data matrix, which we then saved for further analysis. Then, we use the Biobase library package's anyMissing () function to look for missing values in the data matrix. The following R code detects whether there is a missing value or not in the dataset.

```
75  DataMatrixGSE18780 <- as.matrix(NorBgckGSE18780)
76  Biobase::anyMissing(DataMatrixGSE18780)
77
```

Figure 5.9 Sample Code to Detect Missing Values

5.4.6. Screening Outliers

Finding valuable and expressive information is made more accessible by screening for outliers. We used the boxplot () function to identify outliers in expression data and show any expression value in the dataset in these plots. This function helps us visualize normalized data. The following sample code shows box plots for the MA Dataset.

```
79 boxplot(NorBgckGSE18780,
80         xlab="Array Samples",
81         ylab="Normalized Log Intensity",
82         main = "Boxplot of GSE18780 after Normalization and
83         Background Correction ",col=rainbow(6))
84
```

Figure 5.10 Sample Code to Detect Outliers

5.5. High-level Implementation

The following subsections describe the high-level implementation of the MA Data. This subsection presents sample codes using an R programming language to identify DEGs, cluster genes across the samples or cluster samples across genes, and the gene-gene interaction network in detail.

5.5.1. Identifying Differentially Expressed Genes

DEGs show a significant change in their expression level across the samples. The limma library from the Bioconductor package identifies DEGs across samples. Besides this, we followed the steps in GEO2R from the NCBI official website. In the first step, we make proper column names that match the top table of DEGs using the function *make.names()*. The next step is to assign group membership of the samples based on their source. Line 104 depicts this phenomenon. The following sample code can construct proper column names and assign group membership in the dataset. Then the stripsplit() method is used to split the membership number.

```
101 fvarLabels(PF_GSE18780) <- make.names(fvarLabels(PF_GSE18780))
102
103 gmsms <- "000112223313"
104 sml <- strsplit(gmsms, split="")[[1]]
105
```

Figure 5.11 Sample Code to Make Column Names and to Assign Group Membership

The next step in the differential expression analysis is to assign samples to groups and design a matrix for our linear model. After that, we fit into our linear model. The following sample code shows this process.

```
gs <- factor(sml)
groups <- make.names(c("MNInf","MInf","SGNinf","SGInf"))
levels(gs) <- groups
PF_GSE18780$group <- gs
design <- model.matrix(~group + 0, PF_GSE18780)
colnames(design) <- levels(gs)

fit <- lmFit(PF_GSE18780, design)
```

Figure 5.12 Sample Code to Assign Samples to Groups, Design Matrix, and Linear Model Fit

Next, we adjust contrasts of interest and recalculate model coefficients. The following sample code depicts this step.

```
PFGSE18780cts <- paste(groups, c(tail(groups, -1),
                                 head(groups, 1)), sep="-")
cont.matrix <- makeContrasts(contrasts=PFGSE18780cts,
                             levels=design)
PFGSE18780_fit2 <- contrasts.fit(fit, cont.matrix)

```

Figure 5.13 Sample Code to Adjust Contrasts and Recalculate Model Coefficients

The following sample code identifies DEGs. The genes are the top significant genes, and we use them in downstream analysis. We used the *eBayes ()* method to compute the statistics, and this function is Empirical Bayes statistics for differential expression of MA Data. After applying the *eBayes ()* on our data, we extracted the top-ranked genes from the given linear model fit using the *topTable()* function. The following sample code computes statistics and a table of top significant genes from our MA Data.

```
PFGSE18780_fit2 <- eBayes(PFGSE18780_fit2, 0.01)
GSE18780Top_Table <- topTable(PFGSE18780_fit2,
                              adjust="fdr", sort.by="B", number=2500)

```

Figure 5.14 Sample Code to Compute Statistics and Table of Top Significant Genes

We used a function na.omit() from the stats package for data cleaning. The following sample code depicts the data cleaning step for DEGs.

```
CleanGSE18780Top_TableSub <- na.omit(GSE18780Top_TableSub)

```

Figure 5.15 Sample Code for Data Cleaning

The next step in our DEA is visualization and quality control of test results. The differential gene expression analysis assumes that most genes are not DEGs across the samples. The following sample code can build a histogram of P-values for all genes GSE18780.

```
hist(GSE18780TopT2$adj.P.Val, col = "grey",
     border = "white", xlab = "P-adj",
     ylab = "Number of genes",
     main = "P-adj value distribution")

```

Figure 5.16 Histogram of P-Values for all Genes

We can summarize the test results as up, down, or not expressed states and visualize the result using the vennDiagram() function. We can use the function decideTests() from the limma package for the implementation. The following sample code can depict this step.

```
decisionTest <- decideTests(PFGSE18780_fit2,
                            adjust.method="fdr",
                            p.value=0.05)

vennDiagram(decisionTest, circle.col=palette())

```

Figure 5.17 Sample Code to Summarize Test Results of Genes and their Visualization using Venn-diagram

We also visualize our differentially expressed gene using a Q-Q plot for t-statistics after filtering out bad probes. The function plotMD() from the limma package plots the mean-difference of the expression data, and it can highlight statistically significant probes with a p-adj value that is less than 0.05 cutoff. The following sample code shows the mean-difference plot of log fold change vs. the mean log expression value.

```
plotMD(PFGSE18780_fit2, column=GSE18780ct,
       status=decisionTest[,GSE18780ct],
       legend=F, pch=20, cex=1)
abline(h=0)

```

Figure 5.18 Sample Code for Plotting Mean-Difference

5.5.2. Clustering Implementation

This section applies an unsupervised machine learning method on the MA Data to find categories within the data into which the data divide into groups. Before applying the clustering algorithm on the top DEGs, we visualize the data (i.e., the DEGs) using the *heatmap ()* function from the *limma* package. We can use the heatmap and the visualization to learn more about the genes. We selected two sets of genes with three elements in each (i.e., the elements are randomly selected genes) set. Using the matplot () function, a parallel coordinate plot, we can compare gene expression levels between the two sets. Using this sample code, we can look at the genes in the heatmap.

```
set1 <- c("AgaP_AGAP005822", "CLIPB13", "AgaP_AGAP006371")
set2 <- c("AgaP_AGAP002848", "AgaP_AGAP000785", "AgaP_AGAP009906")
matplot(t(GSE18780DEGT[set1, ]),
        type = "l", lwd = 2, col = "skyblue", lty = 1,
        ylim = c(2, 20), xlab = "Samples", ylab = "Expression Values")
for (i in 1:length(set2)) {
    lines(GSE18780DEGT[set2[i], ], type = "l",
          lwd = 2, col = "firebrick")
}

```

Figure 5.19 Sample Code to Study Genes from Heatmap

The main objective of this section is to build a hierarchical clustering of the DEGs, and it is the most basic technique of clustering. The starting point for hierarchical clustering is to measure the proximity(distance) between genes. Sub-section 2.11 gives further detailed information about distance metrics. To calculate the distance, we used the Euclidean distance calculation method. The function *dist()* from the stats package calculates the distance between the genes across the samples. The *hclust()* function clusters the genes with similar expressions using complete linkage hierarchical clustering. The following sample code demonstrates the hierarchical clustering of the unsupervised machine learning approach for the MA Data analysis.

```
DEgdist <- dist(GSE18780DEGT, method = "euclidean")
DEGHclust <- hclust(DEgdist, method = "complete")
plot(DEGHclust)

```

Figure 5.20 Sample Code for Hierarchical Clustering

We can get groups(clusters) from the dendrogram by cutting the tree at some level. The following sample code draws rectangles at different cut levels to get the desired clusters.

```
278 rect.hclust(DEGHclust, k = 2)
279 rect.hclust(DEGHclust, k = 4)
280 rect.hclust(DEGHclust, k = 8)
281 rect.hclust(DEGHclust, k = 10)
282
```

Figure 5.21 Sample Code to Draw Rectangle Cut levels on the Dendrogram

We use the following sample code to retrieve the cluster indices of the genes they belong to and generate parallel coordinate plots of the clusters. The function ***cutree()*** from the stats package cuts the dendrogram tree into a specified number of groups, and in our case, the number of specified groups is four. The ***table()*** function allows us to count the number of occurrences in each cluster. The ***sort()*** function sorts the genes in descending numbers to their clusters(i.e., genes belonging to a group) from the largest to the smallest in our dataset.

```
284 class <- cutree(DEGHclust, k = 4)
285 class
286 table(class)
287 sort(table(class))
288
```

Figure 5.22 Sample Code to Retrieve Cluster Indices of Genes

The ***matplot()*** function visualizes the clustering result and plots them into a parallel coordinate plot. The most important thing here is the data for the specified class. An expression like this gives us a vector of truth or the false value used in the specified object from the DEG expression values. Then, we transposed the top DEGs using the t() function to visualize the clustering results using the ***matplot()*** function. The following sample code can plot four clustering results onto the same screen.

```
291 oPar <- par(mfrow = c(2,2))
292 matplot(t(GSE18780DEGT[class == 4, ]), type = "l", xlab = "Samples", ylab = "Log Expression Values")
293 matplot(t(GSE18780DEGT[class == 3, ]), type = "l", xlab = "Samples", ylab = "Log Expression Values")
294 matplot(t(GSE18780DEGT[class == 1, ]), type = "l", xlab = "Samples", ylab = "Log Expression Values")
295 matplot(t(GSE18780DEGT[class == 2, ]), type = "l", xlab = "Samples", ylab = "Log Expression Values")
296 par(oPar)
297
```

Figure 5.23 Sample Code to Visualize Clustering Results

5.5.2.1. Validating Cluster Technique

We validate the clustering techniques using the clValid() function from the clValid package to determine which clustering technique to use. The function can return results for clustering with validation measures. Among the three validation categories, we use the internal category. Generally, the function helps to determine whether the clustering is good or not based on mathematical sense. The following sample code demonstrates cluster validation.

```
366 validClust <- clValid(GSE18780DEGT, nClust = 2:8,
367                       clMethods = c("hierarchical",
368                                     "kmeans", "diana",
369                                     "fanny", "som", "model",
370                                     "sota", "pam", "clara", "agnes"),
371                       validation = "internal",)
372 summary(validClust)
373 plot(validClust)
374 vignette("clValid")
375
```

Figure 5.24 Sample Code for Cluster Validation

5.5.3. Gene-gene Interaction Network

The discussion in section 4.2.5 discusses the implementation of the gene-gene interaction network using the Cytoscape software. We prepared a dataset for a gene-gene interaction network using the R Studio before loading it into the Cytoscape working environment. We downloaded the stringApp to construct the interaction network and used the up-regulated genes to construct the gene-gene interaction network. We removed those genes that do not have gene symbols and duplicated gene symbols from our dataset. Finally, we exported the dataset in a CSV file format. The following Figure 5.41 depicts the sample code for data preparation implementation to prepare the dataset for a gene-gene interaction network.

```
1384 Cytodata <- up
1385 DataCyto <- subset(GSE18780TopT2, select = c("Gene.symbol"))
1386 Cytodata <- merge(Cytodata, DataCyto, by =0)
1387 rownames(Cytodata) <- Cytodata$Row.names
1388 Cytodata <- Cytodata[, -1]
1389 #Remove Data with Missing Gene Symbol
1390 Cytodata <- Cytodata[!(is.na(Cytodata$Gene.symbol) | Cytodata$Gene.symbol=="") ,]
1391 Cytodata <- Cytodata[!duplicated(Cytodata$Gene.symbol), ]
1392
1393 rownames(Cytodata) <- Cytodata$Gene.symbol
1394 Cytodata <- Cytodata[, -1]
1395 CytoDataExp <- write.csv(Cytodata, "DataForGGI.csv")
1396
```

Figure 5.25 Sample Code to Prepare Data for Gene-gene Interaction Network

The following Figure 5.26 shows the desktop (i.e., the user interface) of the Cytoscape software. We imported the dataset to the working environment from the file. The stringApp plugin can help us select the species, the network type, confidence (score) cutoff, the maximum number of additional interactors, and whether to use intelligent delimiters or/and load enrichment data.

After loading the dataset to the working environment of the stringApp plugin, we copy and paste gene symbols to construct the network. Then we selected the confidence (score) cutoff value to be 0.5 and the maximum number of additional interactors to be the default value (i.e., 0, because the study aims to check the interaction between the given genes only). Adding additional interactors expands the network. Decreasing the confidence (cutoff) score increases the number of nodes (i.e., the number of genes) and vice-versa. The selected network type is a complete STRING network, and the species is Anopheles Gambie. We checked to use both smart delimiters and load enrichment data in the network. This study removes isolated genes (i.e., genes with no interactions).

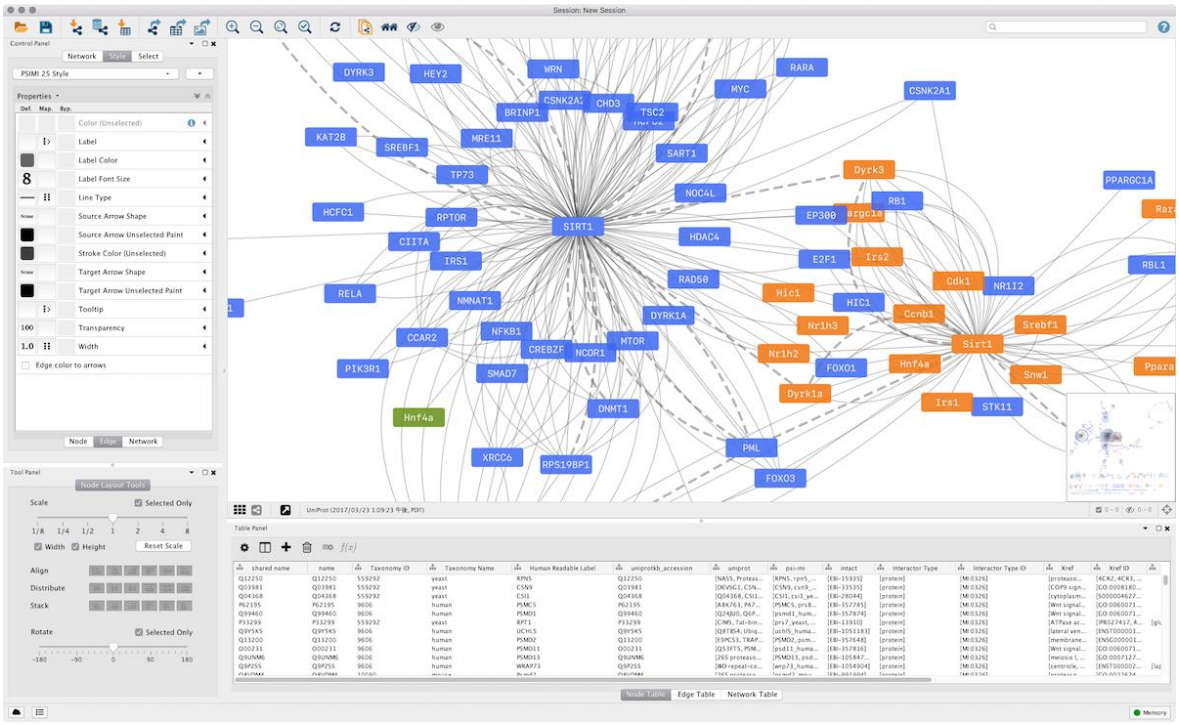


Figure 5.26 Cytoscape User Interface [80]

After constructing the network, we analyze the result using the cytoNCA plugin. We select the essential genes using Eigenvector Centrality calculation using the plugin.

CHAPTER SIX

6. Results, Evaluation, and Discussion

This section discusses the results of the experiments for the analysis of the MA data. The discussion includes the differentially expressed genes, clustering results of the samples and genes, and the gene-gene interaction network as a drug target.

6.1. Preprocessing Implementation Results

This section shows the results of the preprocessing steps in detail. The first step of the implementation was to load the dataset into the working environment in which we got 12 samples of GSE18780 P. falciparum microarray gene expression data. The following figure shows the detail of the dataset.

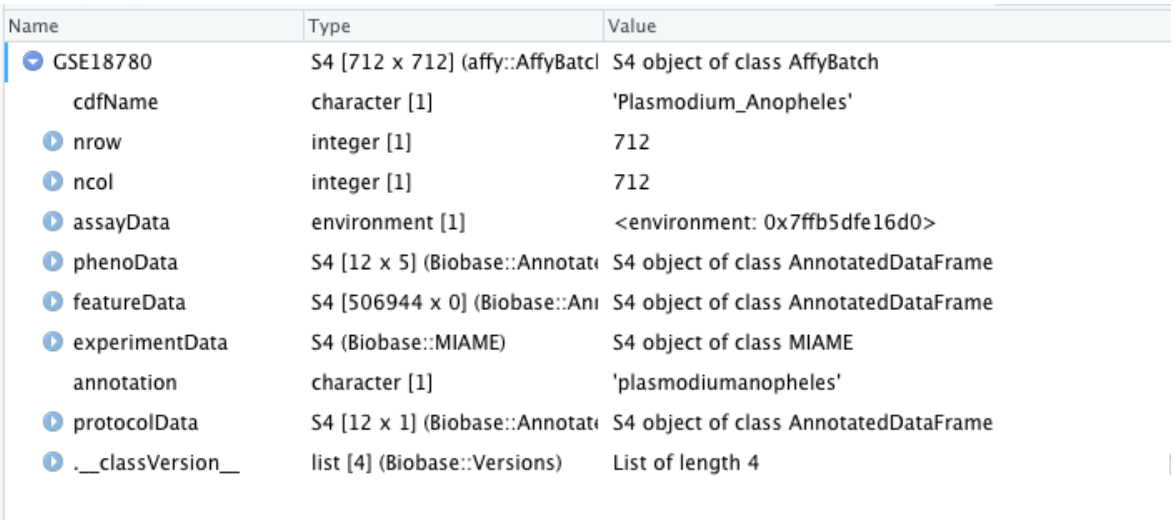
```
AffyBatch object
size of arrays=712x712 features (25 kb)
cdf=Plasmodium_Anopheles (22769 affyids)
number of samples=12
number of genes=22769
annotation=plasmodiumanopheles
notes=
```

Figure 6.1 GSE18780 MA Data Information

From the above figure, we can see that the class of the MA data is an object of AffyBatch. Each of the Plasmodium Anopheles probes has affy ids, and the number of samples is 12 that has 22769 genes in each of the samples (i.e., across the microarrays). We converted this AffyBatch object into an ExpressionSet object in downstream analysis.

6.1.1. Exploring the Dataset Results

Exploring the dataset can help us check whether the MA Data contains all the required components. We can get an additional explanation about the components in chapter five-under section 5.4.2. The following figure shows the components of the microarray dataset.



Name	Type	Value
GSE18780	S4 [712 x 712] (affy::AffyBatcl	S4 object of class AffyBatch
cdfName	character [1]	'Plasmodium_Anopheles'
nrow	integer [1]	712
ncol	integer [1]	712
assayData	environment [1]	<environment: 0x7ffb5dfe16d0>
phenoData	S4 [12 x 5] (Biobase::Annotat	S4 object of class AnnotatedDataFrame
featureData	S4 [506944 x 0] (Biobase::Ani	S4 object of class AnnotatedDataFrame
experimentData	S4 (Biobase::MIAME)	S4 object of class MIAME
annotation	character [1]	'plasmodiumanopheles'
protocolData	S4 [12 x 1] (Biobase::Annotat	S4 object of class AnnotatedDataFrame
.__classVersion__	list [4] (Biobase::Versions)	List of length 4

Figure 6.2 Result for Exploring Microarray Dataset

The raw MA Data (i.e., GSE18780) is an S4 object, a highly complex object type in R used by various Bioconductor packages. The assayData element of the MA Data contains the raw expression value of the genes. The following figure shows the expression values for ten probes across the five samples of the raw MA Data.

	GSM465895	GSM465896	GSM465897	GSM465898	GSM465899
1116_at	2182.540	2669.910	1892.950	164.0190	223.3050
1978_at	307.797	368.902	233.332	15.5215	38.1890
200000_s_at	898.658	398.996	466.798	82.1176	46.6569
200001_at	431.003	1283.570	446.011	25.5867	65.7625
200002_at	2043.790	2377.410	2791.110	210.2200	380.1720
200003_s_at	375.294	972.725	275.327	79.9538	61.6656
200004_at	480.663	1300.210	120.597	78.4208	332.4930
200005_at	327.969	633.450	1577.360	29.9455	34.9027
200006_at	1732.600	2207.320	1467.500	181.9020	236.1000
200007_at	157.049	429.508	214.575	16.4544	44.4383

Figure 6.3 Raw Expression Values for 10 Probes from 5 Samples

6.1.2. Microarray Data Quality Control Results

The MA Data quality control can produce a pdf file for further reading and understanding of the overall MA Data. The file contains five sections in which array comparison, array intensity distribution, variance mean independence, Affymetrix specific plots, and individual array quality are presented and visualized for further understanding.

6.1.3. Background Correction, Normalization, and Data Transformation Result

As discussed in section 5.4.4, we use rma (Robust Multi-Array Average Expression Measure) to perform background correction, normalizing the expression values and transforming the data to log2 scale values. The following diagram shows a boxplot for the raw MA Data before and after applying normalization and background correction techniques on the raw data.

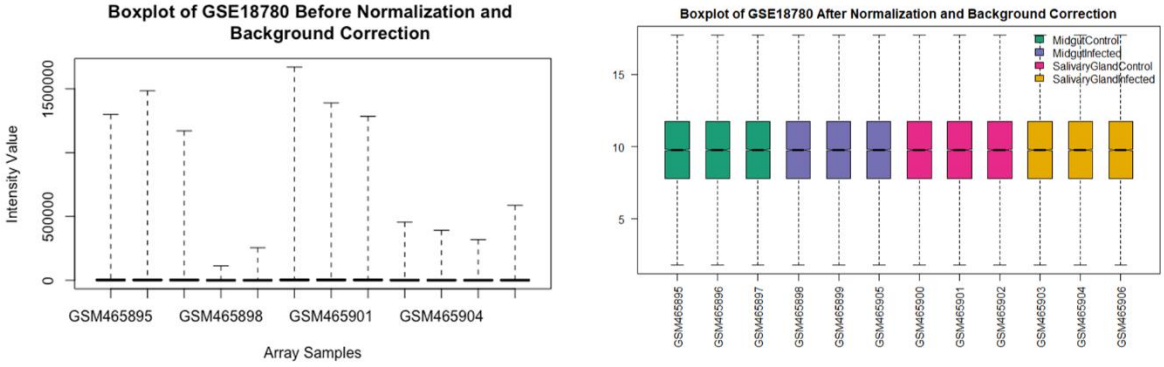


Figure 6.4 Boxplot for Raw GSE18780 MA Data Before and After Normalization, and Background Correction

From Figure 6.4, before applying normalization and background correction, the scale for the arrays' minimum and maximum values is not in some specific number range. Sample number four has the minimum value, whereas sample number 6 has the maximum expression value. In addition to this, the first and the third inter-quartile ranges are not the same for each sample. The median value of each sample is zero. This boxplot can also show that there are no outliers in the dataset.

After applying the normalization and background correction, we can see that the maximum and the minimum values are in the same range. The median value and interquartile range of each sample are the same. Hence this enables us to perform unbiased downstream analysis of the MA Data.

The following Figure shows the histogram for the raw GSE18780 MA data expression value before and after applying log2 transformation.

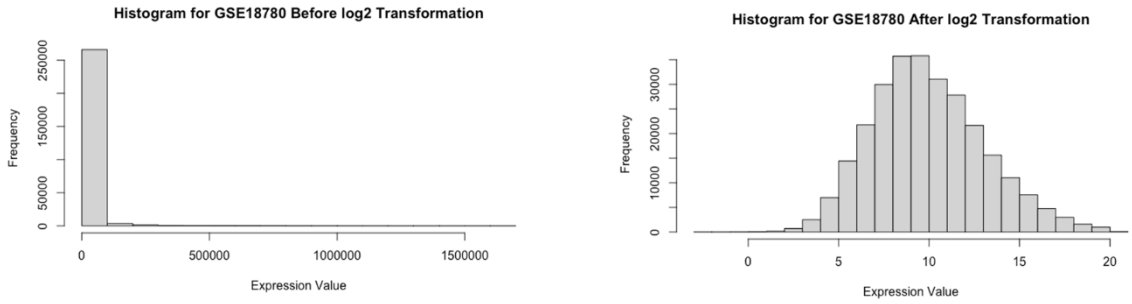


Figure 6.5 Histogram for GSE18780 Before and After log2 Transformation

From the above Figure 6.5, we can see that before log2 transformation, the data is not a normal distribution. The distribution of the values is highly skewed because it has a long right-handed tail. To overcome the skewness, we need to transform the data to have a normal distribution for the downstream analysis, making the data in normal distribution and symmetrical. The right side of the above figure shows the histogram after applying log2 transformation.

We also check for missing values (missing probe ids) and screen outliers after the background correction, normalization, and log scale transformation in the preprocessing. We obtained no missing values and outliers in the preprocessed MA data in the result we obtained. The following figure shows the expression values after preprocessing raw GSE18780 MA data for the first ten probes from 5 samples.

	GSM465895	GSM465896	GSM465897	GSM465898	GSM465899
1116_at	11.091792	11.382575	10.886421	7.357719	7.802872
1978_at	8.265835	8.527094	7.866240	3.956196	5.255085
200000_s_at	9.811628	8.640230	8.866655	6.359620	5.544019
200001_at	8.751554	10.325946	8.800935	4.677322	6.039193
200002_at	10.997031	11.215175	11.446623	7.715756	8.570508
200003_s_at	8.551877	9.925888	8.105002	6.321095	5.946394
200004_at	8.908882	10.344529	6.914050	6.293164	8.377180
200005_at	8.357416	9.307087	10.623296	4.904267	5.125267
200006_at	10.758723	11.108080	10.519145	7.507018	7.883254
200007_at	7.295071	8.746541	7.745338	4.040402	5.473732

Figure 6.6 Preprocessing Result for 10 Probes of 5 samples of GSE18780

6.2. High-level Analysis Results

This section discusses the high-level implementation result of the MA Data, including results for differential expression, clustering of both genes and samples, clustering validation, and finally, gene-gene interaction network.

6.2.1. Differential Expression Result

The differential expression analysis takes the preprocessed data for statistical analysis to discover changes in the level of expression across the microarray samples. In this study, we use all twelve samples to analyze differential expression. Among the 22769 genes, we selected 2500 significant differentially expressed genes according to the False Discovery Rate P-value adjustment from the manually grouped samples of mid-gut control, mid-gut infected, salivary gland control, and salivary gland infected groups. We compared the expression level of the genes in all four groups to extract top-ranked genes using a linear model. Out of 2500 differentially expressed genes, 471 were missing gene symbols, and we removed those genes from the differential expression. The following figure depicts the top 50 differentially expressed genes after data cleaning (i.e., omitting genes that do not have gene symbol names).

```
 [1] "AgaP_AGAP008282" "AgaP_AGAP005822" "AgaP_AGAP002102" "AgaP_AGAP006506"
 [5] "AgaP_AGAP006504" "LYSC4"           "AgaP_AGAP006371" "CLIPB13"
 [9] "AgaP_AGAP006278" "AgaP_AGAP008283" "AgaP_AGAP006421" "AgaP_AGAP008307"
[13] "HPX14"           "AgaP_AGAP000548" "AgaP_AGAP009122" "CLIPB10"
[17] "AgaP_AGAP011514" "AgaP_AGAP006494" "AgaP_AGAP011505" "AgaP_AGAP005712"
[21] "OBP10"           "AgaP_AGAP009859" "AgaP_AGAP007140" "AgaP_AGAP010545"
[25] "AgaP_AGAP007049" "AgaP_AGAP002925" "AgaP_AGAP005890" "AgaP_AGAP003692"
[29] "AgaP_AGAP007851" "AgaP_AGAP008783" "AgaP_AGAP010735" "AgaP_AGAP005256"
[33] "AgaP_AGAP011026" "PPO9"            "AgaP_AGAP004324" "OBP10"
[37] "AgaP_AGAP008013" "AgaP_AGAP003620" "AgaP_AGAP006743" "AgaP_AGAP002353"
[41] "HPX2"            "AgaP_AGAP003606" "AgaP_AGAP010548" "AgaP_AGAP005889"
[45] "AgaP_AGAP011317" "AgaP_AGAP006504" "AgaP_AGAP005712" "AgaP_AGAP003248"
[49] "AgaP_AGAP003977" "AgaP_AGAP005693"
```

Figure 6.7 Top 50 Differentially Expressed Gene Symbols

The following Venn diagram based on p.value=0.05 depicts the number of DEGs across the samples. From the diagram, we can conclude that overlapping genes in each of the four manually grouped gene groups need to be solved using the classification machine learning approach. One hundred ninety-five genes between midgut control and midgut infected, 2567

genes between midgut infected and salivary gland control, 27 genes between salivary gland control and salivary gland infected and 1091 genes between salivary gland infected and midgut control are DEGs. The rest of 4298 genes expression is common to all groups, and 14591 genes are not DEGs.

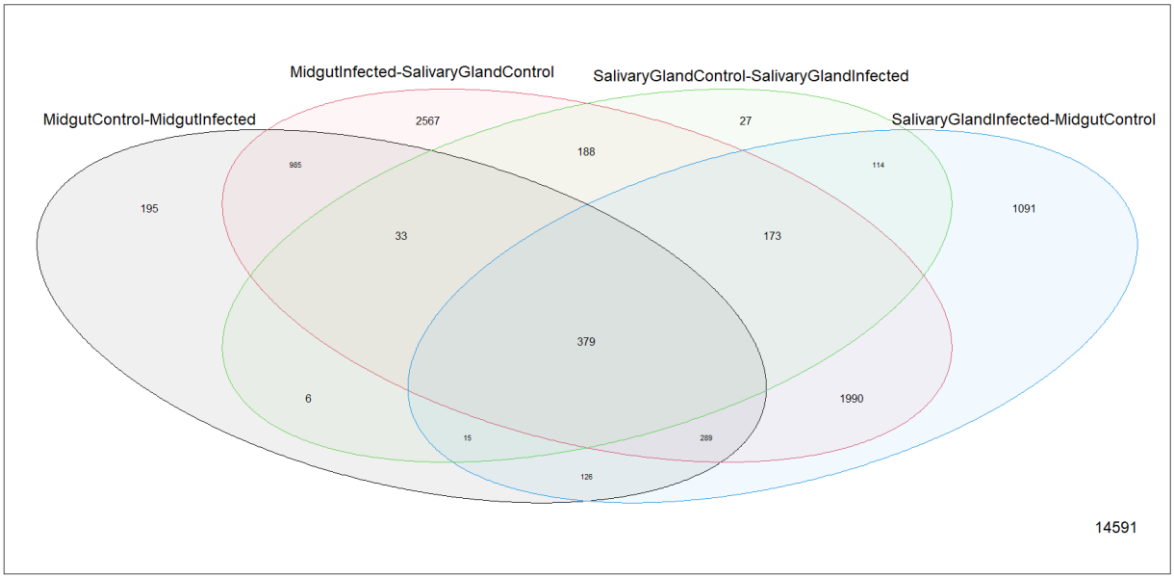


Figure 6.8 Venn Diagram of Differentially Expressed Genes

The following mean-difference plot depicts log-intensity ratios (log-fold change) versus log-intensity averages (average log expression) values for samples under mid-gut infected versus the salivary gland control group. The red points are genes with high expression (i.e., up-regulated genes) and significant ones, whereas the blue points have a low expression (i.e., downregulated ones) and significant ones. The black points are not DEGs, and they are not significant.

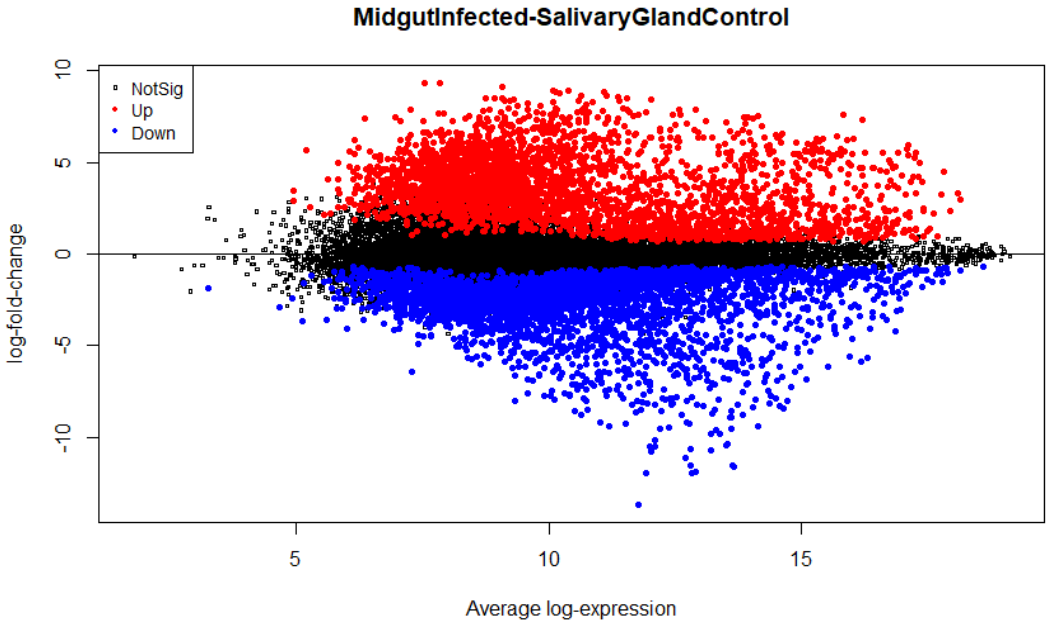


Figure 6.9 Mean-Difference Plot for Midgut Infected Vs. Salivary Gland Control Groups

6.2.2. Clustering Result

This section discusses P. falciparum MA data's clustering result for samples and genes. The expression value of each gene is an independent element. The first step in the clustering analysis of the P. falciparum MA data is to visualize the data using a heatmap. Heatmap can help us to study the clustering of the genes intuitively. We plot 25 DEGs across the whole samples in the heatmap for clear visualization. The structure is the same if we replot the heatmap for all DEGs.

From the following Figure 6.10, we can intuitively explain that the red, yellow, and white colors are the expression levels of the genes. The heatmap represents the expression levels into a spectrum of colors corresponding to warmer and cooler colors. Probably red values would be high expression levels, whereas yellow or white values are low expression levels. Rearrangement of the columns cannot affect the result because the information is in the topology of the diagram, not in the ordering.

The lines at the top of the diagram can represent the similarity relationships between samples. In contrast, the left side of the diagram can represent the similarity between genes. The lettering on the right side of the diagram represents gene symbols. We can see a clear similarity structure among the samples and the genes. The heatmap can also show samples

GSM465895, GSM465897, and GSM465896 are similar, GSM465905, GSM465899, and GSM465898 are similar, GSM465900, GSM465902, and GSM465901 are similar, and finally, GSM465904, GSM465903, and GSM465906 are similar. The weakness of this diagram representation is that it cannot show the degree of similarity among the samples and the genes, and it depends on the details of the algorithm. If we use the algorithm with different parameters, the result may not be like Figure 6.10. So, we are just looking at what the diagram shows us in principle.

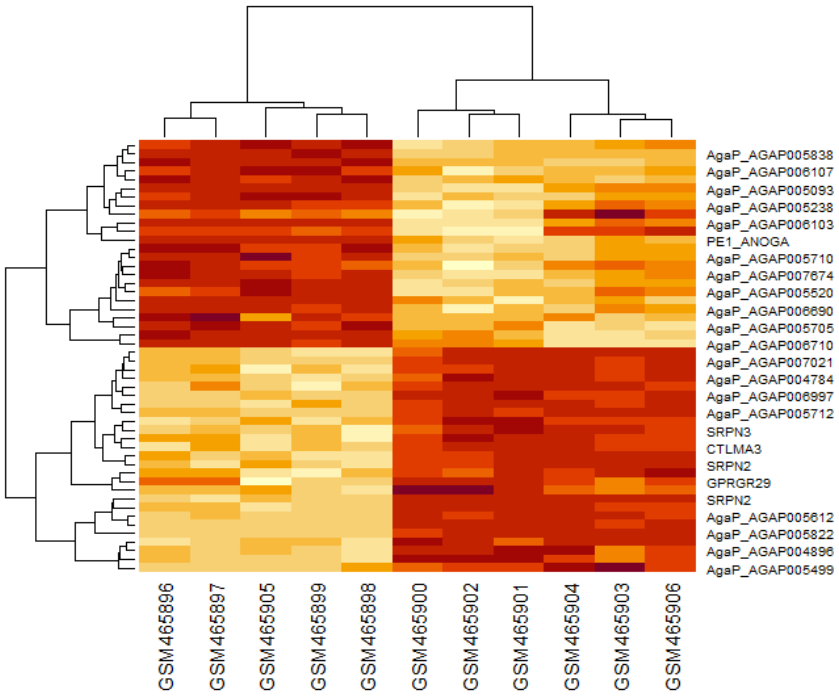


Figure 6.10 Heatmap of 45 Differentially Expressed Genes

From the above Figure 6.10, we can see that there seem to be many highly expressed genes in samples; GSM465900, GSM465902, GSM465904, GSM465903, GSM465906, and GSM465901 in comparison to GSM465900, GSM465902, and GSM465901 samples. The Figure depicts that there seem to be many lowly expressed genes in samples; GSM465896, GSM465897, GSM465895, GSM465905, GSM465899, and GSM465898.

The following matplot Figure 6.11, shows randomly selected genes with similar expression patterns into two sets (i.e., three randomly selected genes are in each set). The genes in set1 (i.e., "AgaP_AGAP005822", "CLIPB13" and "AgaP_AGAP006371") are colored in light-blue whereas the genes in set2 (i.e., "AgaP_AGAP002848", "AgaP_AGAP000785", "AgaP_AGAP009906") are colored in red. The diagram shows the genes according to their

index in the set (i.e., the first line in the set1 is the gene at the first index, the second line is the representation of the gene at index two, and lastly, the third line represents the gene at index three in the set).

The X-lab of the figure represents the samples in our microarray dataset, whereas the Y-lab represents the expression value of the genes. The genes in set1 are lowly expressed in the first five samples, whereas set2 contains high expression values in the first five samples. Then the genes in set1 are highly expressed starting from the sixth sample up to the tenth sample, and the reverse is true for genes in set2 as well. Lastly, the genes in set1 are lowly expressed in sample number eleven and highly expressed in sample number twelve, and still, the rivers are true for genes in set2. Gene set1 and set2 are very in opposite ways.

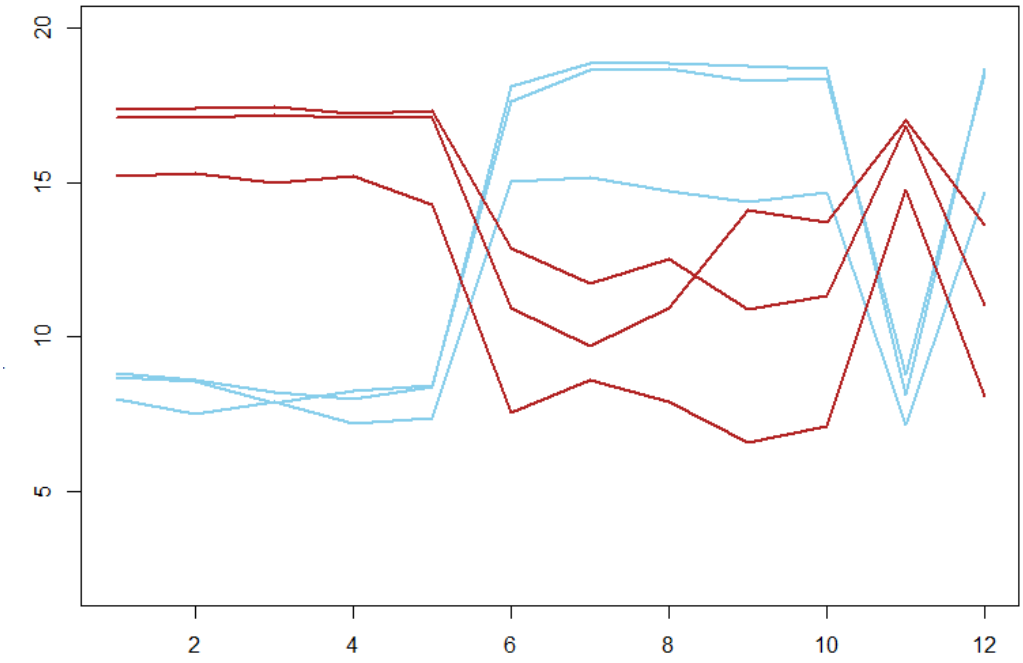


Figure 6.11 matplot of Randomly Selected Genes

The next big thing in this section is to depict the hierarchical clustering of the differentially expressed genes. Hierarchical clustering has the advantage of being simple and allowing for easy visualization and interpretation of the results. The clustering distance matrix produces a dendrogram that connects the most similar genes. Then those connected groups of genes are added together to produce more significant connected genes. The following Figure 6.12 depicts the hierarchical clustering of GSE18780 P. falciparum MA data informative genes. To visualize clearly, we plot the hierarchical clustering result for 40 randomly selected genes of the MA Data in

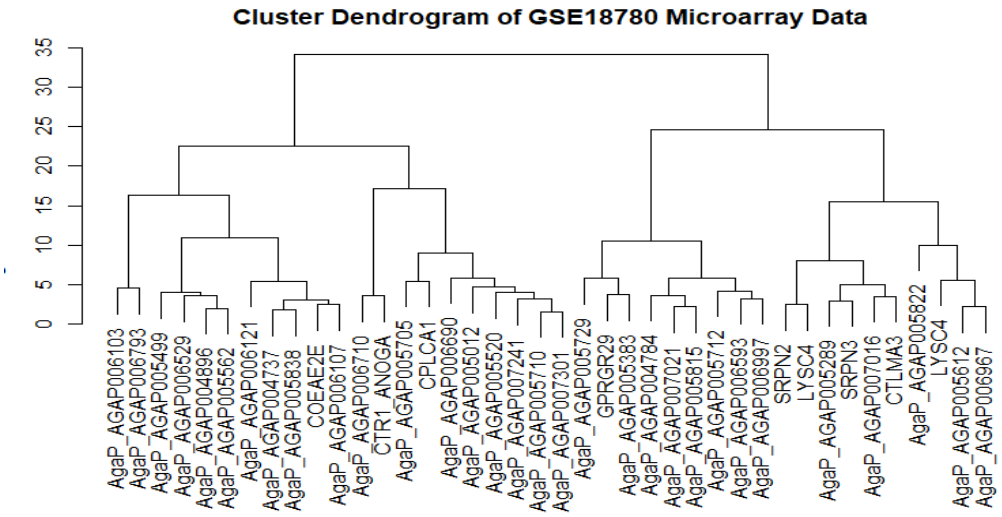


Figure 6.12 Hierarchical Clustering of 40 Randomly Selected DEG

The following Figures show the result of using different proximity measuring methods in measuring the distance and the effect on the visualization of the results.

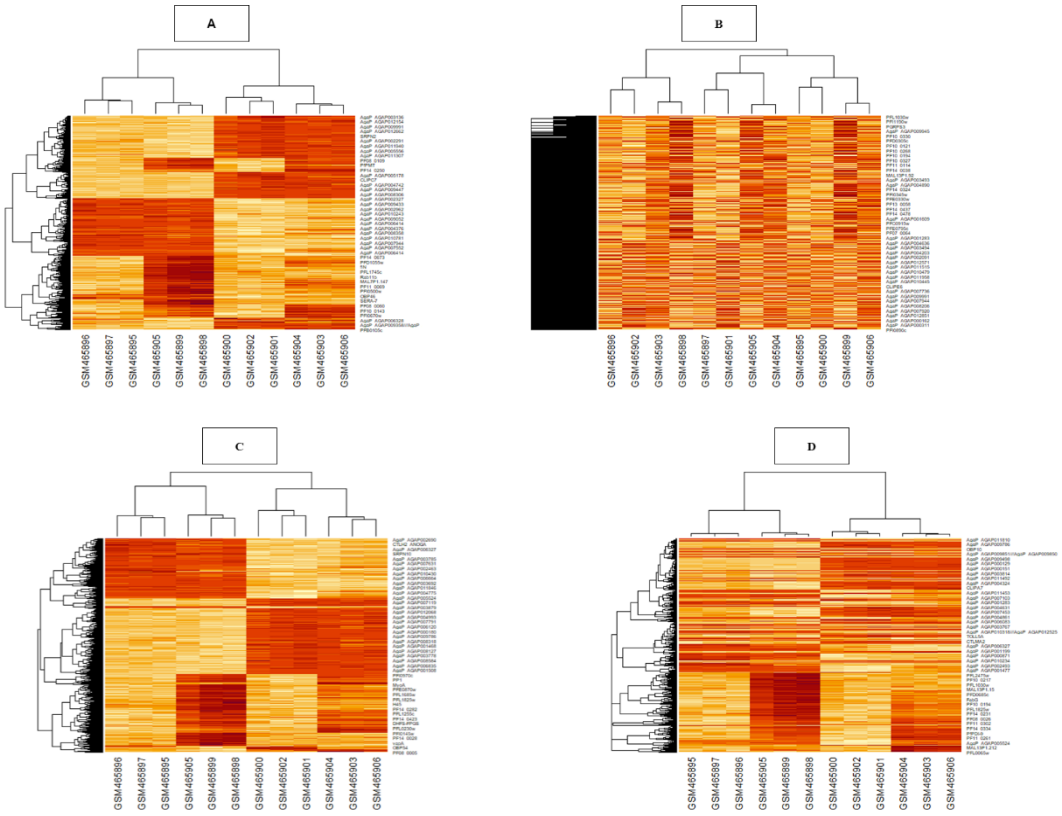


Figure 6.13 **A**: Heatmap Using Canberra Proximity Measure, **B**: Heatmap Using Maximum Information Gain Proximity Measure, **C**: Heatmap Using Maximum Distance Proximity Measure, **D**: Heatmap Using Pearson Correlation Proximity Measure

From Figure 6.13, we can conclude that we can get a clear structure in the heatmap using Canberra and Maximum Distance proximity measures. The clusters are clearer and interpretable intuitively. In the case of the Maximum Information Gain and Pearson Correlation proximity measure, the number of clusters is overwhelming and hard to interpret intuitively. There is no clear cluster in both cases.

To get clusters from the heatmap, we cut the dendrogram at different levels (i.e., drawing rectangles at a different level). Cutting the dendrogram makes it fall apart into small parts, and we can treat each of the smaller parts as a cluster. There is no precise underlying mechanism to cluster the genes into one, two, or many groups in biological data, and it is due to that the difference between the genes is continuous. There are genes expressed in one way, genes expressed in another, and genes in the middle of those, making it difficult to put the genes in the cluster. First, we cut the dendrogram into two. Then we cut the dendrogram into four, eight, and ten. Figure 6.15 depicts cutting the dendrogram into four to get the clusters.

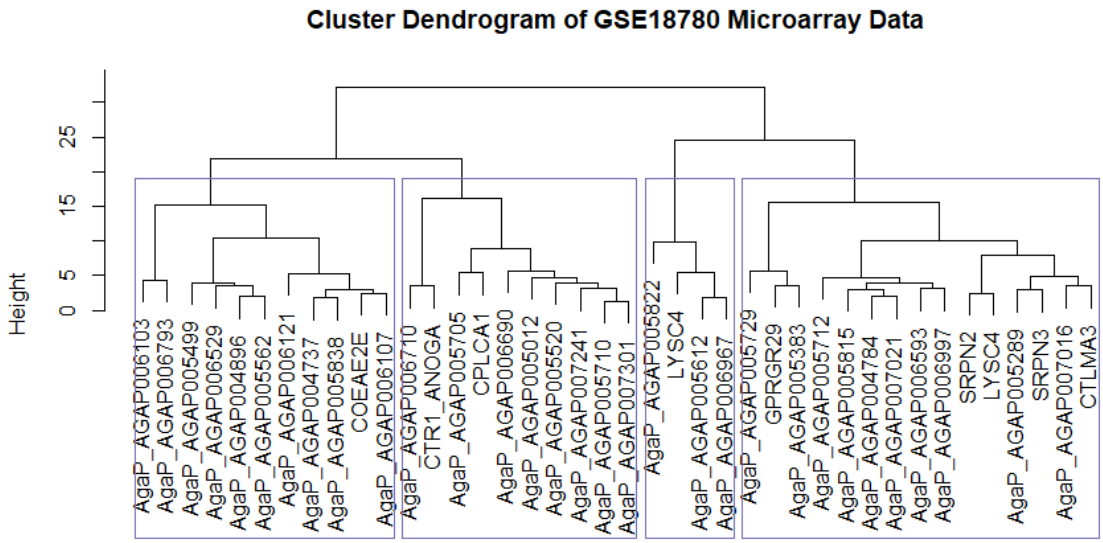


Figure 6.14 Cutting the Dendrogram of 40 Randomly Selected DEG into Four to get Four Clusters

This study cut the tree into four, resulting in 226 genes falling in the fourth cluster, 283 genes in the third cluster, 616 genes in the first cluster, and 904 genes in the second cluster. The following matplot shows the four most enormous clustering results using the Euclidean

distance proximity measure and complete linkage. In Figure 6.15, the X-coordinate is the number of samples, whereas the Y-coordinate is the expression value of the genes. If we look at the individual plots, we can see those genes with similar expression patterns. There is no similarity among the four plots. The scale is similar for plots in the first row, and the same is true for plots in the second row. Some of the genes are shallow in the first row of the first plot. There is a significant difference between all four plots, demonstrating that the similarity between clusters decreases. Clusters look similar in our naked eye but appear to be sufficiently dissimilar in the mathematical sense. The clusters are not homogenous, making them fit for biological interpretation.

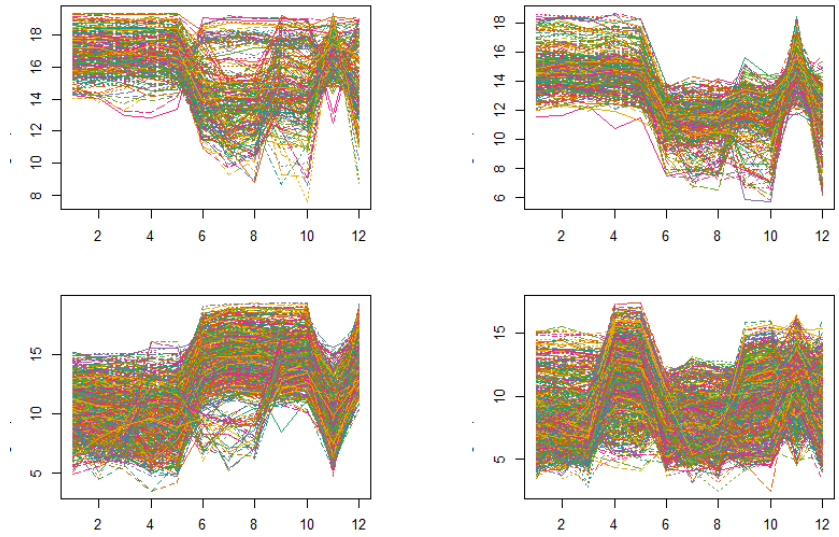


Figure 6.15 Clustering Result of GSE18780 Differentially Expressed Genes matplot

6.2.3. Clustering Model Validation

This section discusses the validation measure for a given set of clustering algorithms. We employed an internal validation measure, as mentioned in section 3.4.3.1, to validate our clustering approach. The internal validation measure depends on the information in the specified data only, which measures the clusters' features, such as compactness within clusters and separation among clusters. In real-world scenarios, we want our clustering results to be as compact and separated as possible. Figure 6.16 is a summary output taken from the R studios console.

```
Cluster sizes:
 2 3 4 5 6 7 8

Validation Measures:
                                 2         3         4         5         6         7         8

hierarchical Connectivity    65.8508   73.9175   89.7397   99.2036  138.2722  155.0214  158.5083
             Dunn             0.0363    0.0473    0.0473    0.0473    0.0473    0.0433    0.0433
             Silhouette       0.3404    0.3836    0.3104    0.2763    0.2253    0.1993    0.1801
kmeans       Connectivity   135.0885  127.2496  252.6274  347.4091  374.2786  395.3698  439.9179
             Dunn             0.0321    0.0376    0.0431    0.0367    0.0372    0.0381    0.0392
             Silhouette       0.3625    0.3957    0.3107    0.3041    0.2981    0.2863    0.2795
diana        Connectivity   121.8393  140.9972  194.7135  252.3409  252.3409  385.1798  442.1056
             Dunn             0.0288    0.0260    0.0294    0.0310    0.0310    0.0325    0.0325
             Silhouette       0.3626    0.3647    0.3341    0.3012    0.2845    0.2628    0.2465
fanny        Connectivity   127.2433  225.2921  349.4048  312.3821  358.4341        NA        NA
             Dunn             0.0284    0.0262    0.0342    0.0284    0.0284        NA        NA
             Silhouette       0.3619    0.3261    0.2749    0.1669    0.1662        NA        NA
som          Connectivity   137.1905  132.8175  184.0306  294.7151  369.2663  435.6552  471.7516
             Dunn             0.0315    0.0301    0.0338    0.0367    0.0421    0.0383    0.0421
             Silhouette       0.3614    0.3968    0.3599    0.3239    0.2974    0.2875    0.2708
model        Connectivity   300.1333  338.9175  638.6841 1060.6365 1218.7790 1054.7560 1212.0151
             Dunn             0.0209    0.0226    0.0198    0.0229    0.0208    0.0231    0.0235
             Silhouette       0.2711    0.3572    0.2565    0.1739    0.1478    0.1453    0.1139
sota         Connectivity   196.0683  320.8675  383.0837  429.6345  548.8190  563.5917  579.1726
             Dunn             0.0297    0.0170    0.0170    0.0215    0.0229    0.0229    0.0229
             Silhouette       0.2987    0.2858    0.2217    0.2368    0.2399    0.2280    0.2018
pam          Connectivity   154.0754  136.8853  169.6647  273.2937  403.0512  416.5016  497.8909
             Dunn             0.0288    0.0301    0.0365    0.0289    0.0340    0.0344    0.0379
             Silhouette       0.3605    0.3972    0.3576    0.3228    0.2966    0.2819    0.2638
clara        Connectivity   128.5040  193.5159  274.6563  306.2504  409.4540  490.9139  571.1810
             Dunn             0.0409    0.0295    0.0324    0.0201    0.0314    0.0323    0.0351
             Silhouette       0.3557    0.3891    0.3479    0.3175    0.2718    0.2626    0.2430
agnes        Connectivity    65.8508   73.9175   89.7397   99.2036  138.2722  155.0214  158.5083
             Dunn             0.0363    0.0473    0.0473    0.0473    0.0473    0.0433    0.0433
             Silhouette       0.3404    0.3836    0.3104    0.2763    0.2253    0.1993    0.1801

Optimal Scores:

             Score   Method       Clusters
Connectivity 65.8508 hierarchical 2
Dunn          0.0473 hierarchical 3
Silhouette    0.3972 pam          3
```

Figure 6.16 Summary Statement of All the Validation Measures

The above Figure 6.16 shows that Hierarchical clustering with 2 and 3 clusters performs the best in Connectivity and Dunn index, whereas pam clustering with 3 clusters performs the best in silhouette. The validation measures are displayed graphically in the following Figure 6.17. Bear in mind that silhouette width and Dunn index should be maximum, whereas the connectivity should be minim. The figure clearly shows that hierarchical clustering outperforms the other clustering methods under the connectivity and Dunn index validation measures for all of the clustering methods evaluated. In contrast, pam outperforms all clustering methods evaluated under silhouette validation measure. The optimal number of clusters seems to be either two or six using connectivity and Dunn index regardless of the clustering algorithms. The number of clusters is three for the silhouette width, even if it seems less clear.

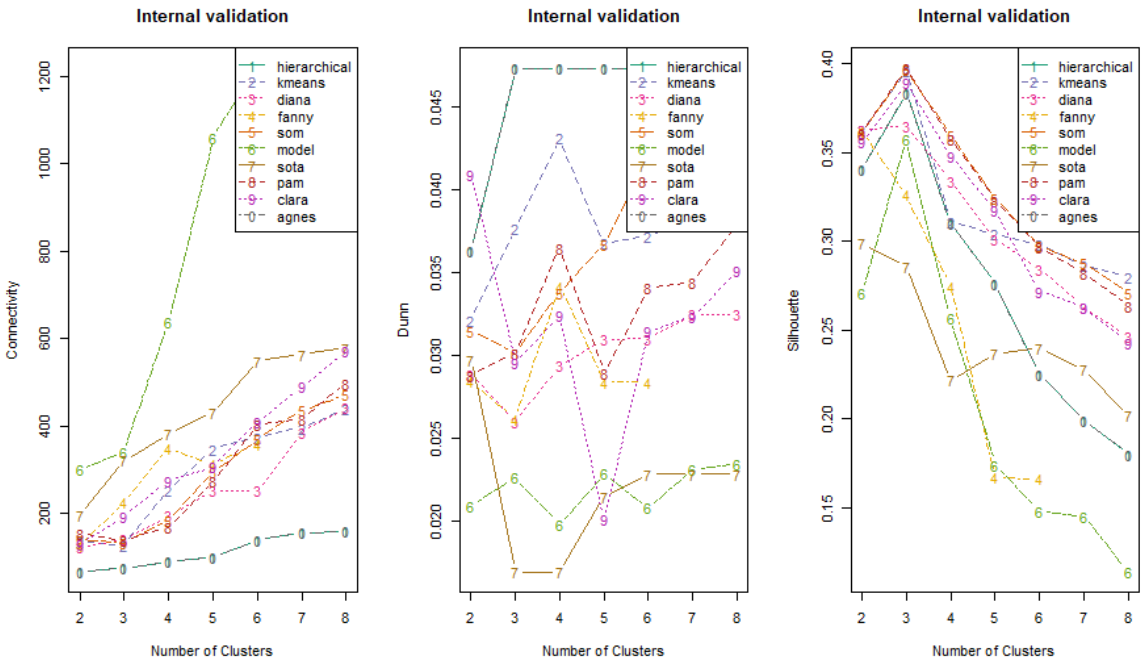


Figure 6.17 Plots of the Connectivity Measure, the Dunn Index, and the Silhouette Width

6.2.4. Gene-Gene Interaction Network Result

This section discusses the gene-gene interaction network result for the up-regulated 1316 genes. Removal of genes with duplicated and missing gene symbols results in 1038 genes. We loaded those 1038 genes to the working environment of the Cytoscape to construct the gene-gene interaction network. After removing isolated genes (gene with no interaction), we visualized the following gene-gene interaction network as shown in Figure 6.19. In the figure, thick lines represent very high confidence in the interaction between the genes (nodes).

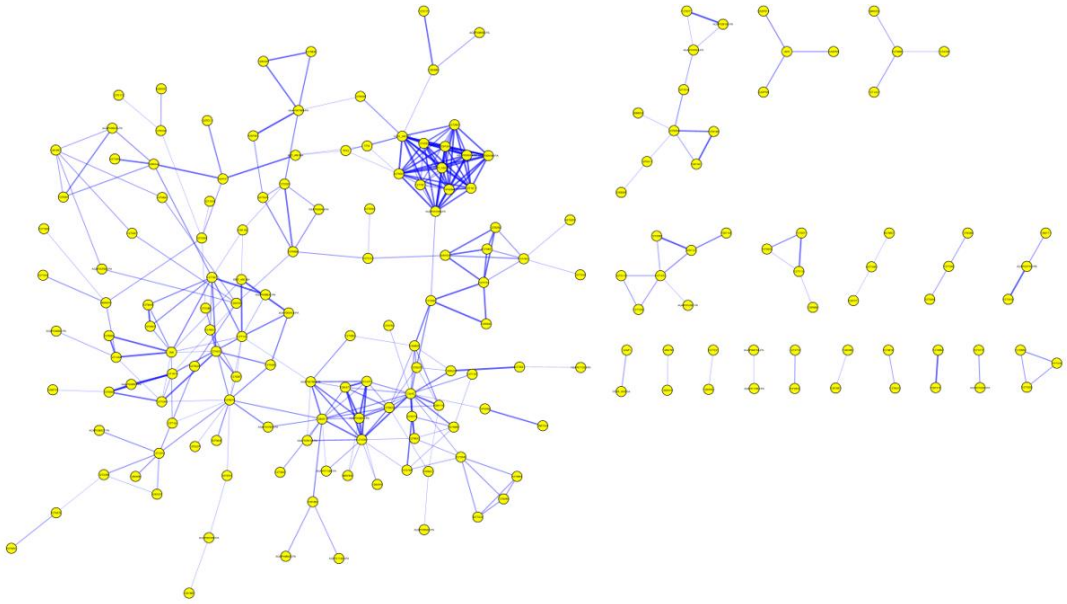


Figure 6.18 Gene-gene Interaction Network for Up-regulated Genes

Figure 6.48 shows the top 20 essential genes used as potential therapeutic targets. We can access the genes using their names.

No.	Name	Degree	Eigenvector	Betweenness	Closeness
1	7165.AGAP003153-PB	15.0	0.30659133195877075	1966.9965236517878	0.0027476661557753936
2	7165.AGAP002401-PA	13.0	0.30170395970344543	399.0324231324232	0.0027473292950007523
3	7165.AGAP010298-PA	12.0	0.29442793130874634	2651.8978501031143	0.0027481256445274394
4	7165.AGAP003879-PA	11.0	0.29197415709495544	0.4444444444444444	0.00274688537743208
5	7165.AGAP009334-PA	11.0	0.29197415709495544	0.4444444444444444	0.00274688537743208
6	7165.AGAP002884-PA	11.0	0.2919740378856659	0.4444444444444444	0.00274688537743208
7	7165.AGAP003430-PA	11.0	0.2919740378856659	0.4444444444444444	0.00274688537743208
8	7165.AGAP003588-PA	11.0	0.2919740378856659	0.4444444444444444	0.00274688537743208
9	7165.AGAP002473-PA	11.0	0.2919740378856659	0.4444444444444444	0.00274688537743208
10	7165.AGAP001823-PA	10.0	0.27111271023750305	0.0	0.002746870072488397
11	7165.AGAP009486-PA	10.0	0.27111268043518066	0.0	0.002746870072488397
12	7165.AGAP006389-PA	9.0	0.24599036574363708	0.0	0.0027468547677152633
13	7165.AGAP011054-PA	3.0	0.057359132915735245	449.159718059718	0.002746931293286455
14	7165.AGAP000396-PA	3.0	0.057359132915735245	449.159718059718	0.002746931293286455
15	7165.AGAP011754-PA	3.0	0.028905237093567848	518.0	0.002745737979738347
16	7165.AGAP010687-PA	5.0	0.028770679607987404	4102.046138586442	0.0027490297542043986
17	7165.AGAP005929-PA	2.0	0.028683319687843323	832.2519793072423	0.0027468394631126763
18	7165.AGAP010404-PA	3.0	0.010738966055214405	995.8880674880675	0.0027468394631126763
19	7165.AGAP002440-PC	16.0	0.003958225250244141	5152.633537406041	0.0027497197293784963
20	7165.AGAP012014-PC	8.0	0.003755025565624237	622.5032397753843	0.002748861146269522

Figure 6.19 Top 20 Essential Genes which are Potential Therapeutic Target

6.3. Discussions

The differential expression analysis was the start of the high-level analysis. We selected 2500 genes as differentially expressed genes according to the P-value adjustment of the false discovery rate. We used the P-value to lower the number of genes to 2500 from 22769 genes. We used the Bayes linear model to rank the genes according to their expression level across the samples. We can access genes using their gene symbol; in this study, we only consider

genes with gene symbols, resulting in the removal of 471 genes from the 2500 genes. Some of the genes from the MA Data are DEGs in the same group or sample, resulting in an overlapping number of genes in each group. This problem can be solved using a supervised machine learning approach.

The unsupervised machine learning approach comes next to the differential expression analysis. This unsupervised machine learning results in the clustering of both genes and samples based on the interest of the study. Clustering helps to study the genes or the samples by grouping genes with the same behavior or samples with the same behavior. Beyond matplot, we used heatmaps as a visualization tool for studying the clustering result.

The first unsupervised machine learning technique used in this study is hierarchical clustering, which can group similar genes. Vilda Purutc uo glu Gazi and Elif Kays[58] used the hierarchical clustering technique to compare clustering techniques on actual MA data. This study applies five distance measuring methods (i.e., Euclidian, Canberra, Maximum Information Gain, Maximum Distance Measure, and Pearson Correlation). The Canberra and Maximum distance measures produce the best results in this study. The advantage of hierarchical clustering is that it is simple to read and understand. However, the disadvantage is that it is difficult to return to the tree's branches if a single error exists in the branch of the tree.

This study addresses the stated question based on the results obtained and validation metrics. The first research question is: *How to identify Differentially Expressed Plasmodium Falciparum Anopheles Gambie mosquito genes?* This study performs the differential gene expression analysis using the limma package from Bioconductor to answer the question. Using the P-values and LFC from the manually grouped samples of mid-gut control, mid-gut infected, salivary gland control, and salivary gland infected, we selected 2500 genes that showed significant differential expression among the 22769 total genes. Genes with missing gene symbols were deleted from the differential expression analysis of 2500 genes. A total of 195 genes, 2567 genes, 27 genes, and 1091 genes are DEGs between midgut control and infected, infected and salivary gland control, and infected and midgut control, respectively. The expression of 4298 genes is shared by all groups, while 14591 genes are not differentially expressed (DEGs).

The second research question is *How to identify P. Falciparum Anopheles Gambie genes with the same biological behavior?* This study performs the clustering of genes across the samples using an unsupervised machine learning technique to answer this question. We applied hierarchical clustering technique. Using this method, the tree was divided into four clusters, with 226 genes falling into the fourth cluster, 283 genes falling into the third cluster, 616 genes falling into the first cluster, and 904 genes falling into the second cluster according to the findings. The matplot depicts the four most massive clustering results obtained using the Euclidean distance proximity metric and complete linkage. Internal validation is applied in this study. The validation compares different clustering techniques in the given microarray dataset. It gives us hierarchical clustering as the best based on our question of interest. The following best is the pam clustering technique for this research. Hierarchical clustering with 2 and 3 clusters performs the best in Connectivity and Dunn index, whereas pam clustering with 3 clusters performs the best in silhouette.

The third research question is *Which of the Plasmodium Falciparum Anopheles Gambie genes are potential drug targets?* To answer this question, we construct the gene-gene interaction network of the upregulated genes. We selected a list of essential genes based on eigenvector centrality from the network. The maximum eigenvector is ~0.3066, and the minimum eigenvector is 0.0. We removed isolated genes (i.e., genes that do not interact). The essential genes can be potential drug targets for developing drugs that can alter the production of proteins from those genes.

CHAPTER SEVEN

7. Conclusion, Recommendation, and Future Work

This chapter includes the conclusion of the P. falciparum MA Data analysis, the recommendation, and future work for additional and valuable tasks.

7.1. Conclusion

This research proposed to analyze P. Falciparum MA Data using machine learning approaches to find potential drug targets from the data. To accomplish the study, we perform preprocessing first and then high-level analysis steps. In doing so, we collected the MA Data from public repositories and performed preprocessing analysis to make it available and applicable for downstream analysis (i.e., high-level analysis). In the low-level analysis mainly, we used the limma and other packages from Bioconductor in our study. We successfully applied all the essential preprocessing steps to implement the preprocessing. The preprocessing step includes MA Data quality control, background correction, normalization, data transformation, missing value imputation, and dealing with outliers. We validated the preprocessing result using visualization techniques.

The high-level analysis starts with finding DEGs across samples. For this, we used 0.01 (a numeric value found between 0 and 1), assuming the proportion of DEGs. The DE result gives us 2500 genes adjusted by FDR and sorted using the character B (i.e., it is a character string that can specify the statistic to rank the genes by possible values in the top table). Clustering follows the DEG identification in which we cluster genes based on their expression values. A parallel coordinate plot visualizes the actual expression level of the genes. We applied Hierarchical clustering techniques in the dataset. Validation of the clustering technique helps select the clustering technique to use in the dataset. The proposed study uses the internal validation technique to determine whether the clustering technique is suitable based on mathematical sense. This validation technique suggests that hierarchical clustering is the best followed by the pam clustering technique.

The last step in the high-level analysis is constructing a gene-gene interaction network to show potential drug targets. This study visualizes the network using the Cytoscape software with a cutoff value of 0.4. The gene-gene interaction network shows the connection between

genes that can produce proteins, and then it shows the interaction of one gene with the other genes. This study designs the gene-gene interaction network for the upregulated genes only.

7.2. Recommendation

This study identifies potential drug targets; therefore, we recommend that pharmacologists use this study to develop new drugs or vaccines for malaria disease. In addition to pharmacologists, we recommend NGOs working on malaria diseases to use this research to support developing new drugs and vaccines. The other recommendation is for researchers interested in working with MA Data analysis, and this research can be used as a blueprint or as an input for their research.

7.3. Future Work

This study uses MA Data to analyze P. Falciparum gene expression values using machine learning approaches; we suggest using RNAseq data in future work. The study only focuses on visualizing gene-gene interaction networks using the Cytoscape software. We suggest that researchers in the future work on the detailed analysis of the gene-gene interaction network and the functional enrichment analysis using machine learning and deep learning techniques. We also suggest that researchers work on other species of the Plasmodium parasite family's microarray and RNAseq data in the future for better understanding and comparison of the parasite to deal with the diseases it causes. This study does not provide a solution for the significant overlapping of genes between groups or clusters while searching for differentially expressed genes. The study leaves this problem as a future work to be solved using the classification machine learning technique.

This research was funded by Adama Science and Technology University with a grant number ASTU/SM-R/080/19.

Reference:

[1] F. E. Cox, “History of the discovery of the malaria parasites and their vectors,” *Parasites and Vectors*, vol. 3, no. 1, pp. 1–9, 2010, doi: 10.1186/1756-3305-3-5.

[2] *World malaria report 2019*. Geneva: World Health Organization; 2019. Licence: CC BY-NC-SA 3.0 IGO.

[3] B. Xiao *et al.*, “Epigenetic editing by CRISPR/dCas9 in Plasmodium falciparum,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 116, no. 1, pp. 255–260, 2019, doi: 10.1073/pnas.1813542116.

[4] K. N. Olafson, T. Q. Nguyen, J. D. Rimer, and P. G. Vekilov, “Antimalarials inhibit hematin crystallization by unique drug–surface site interactions,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 114, no. 29, pp. 7531–7536, 2017, doi: 10.1073/pnas.1700125114.

[5] “Malaria.” https://www.who.int/news-room/fact-sheets/detail/malaria (accessed Jan. 18, 2022).

[6] L. S. Garcia, “Malaria,” *Clin. Lab. Med.*, vol. 30, no. 1, pp. 93–129, 2010, doi: 10.1016/j.cll.2009.10.001.

[7] Q. Jing *et al.*, “Plasmodium falciparumvar Gene Is Activated by Its Antisense Long Noncoding RNA,” *Front. Microbiol.*, vol. 9, no. December, pp. 1–9, 2018, doi: 10.3389/fmicb.2018.03117.

[8] Z. Bozdech *et al.*, “The transcriptome of the intraerythrocytic developmental cycle of Plasmodium falciparum,” *PLoS Biol.*, vol. 1, no. 1, pp. 85–100, 2003, doi: 10.1371/journal.pbio.0000005.

[9] M. Molla, M. Waddell, D. Page, and J. Shavlik, “USING MACHINE LEARNING TO DESIGN AND INTERPRET GENE-EXPRESSION MICROARRAYS.”

[10] H. El-Shishiny, T. H. Soliman, and M. El-Asmar, “Mining drug targets based on microarray experiments,” *Proc. - IEEE Symp. Comput. Commun.*, pp. 175–181, 2008, doi: 10.1109/ISCC.2008.4625615.

[11] A. Mohammadi, M. H. Saraee, and M. Salehi, “Identification of disease-causing genes using microarray data mining and Gene Ontology,” *BMC Med. Genomics*, vol. 4, no. 1, p. 12, 2011, doi: 10.1186/1755-8794-4-12.

[12] N. K. Shah and N. Valecha, “Antimalarial drug resistance,” *Recent Adv. Malar.*, pp. 383–407, 2016, doi: 10.1002/9781118493816.ch14.

[13] S. Saviozzi, G. Iazzetti, E. Caserta, A. Guffanti, and R. A. Calogero, “Microarray data analysis and mining.,” *Methods Mol. Med.*, vol. 94, no. 3, pp. 67–90, 2004, doi:

10.1385/1-59259-679-7:67.

[14] K. Baggerly and B. Broom, “Analysis of Microarray Data Lecture 2 : The Basics of dChip,” *Cancer*, 2010.

[15] D. B. Allison, X. Cui, G. P. Page, and M. Sabripour, “Microarray data analysis: From disarray to consolidation and consensus,” *Nat. Rev. Genet.*, vol. 7, no. 1, pp. 55–65, 2006, doi: 10.1038/nrg1749.

[16] F. Cordero, M. Botta, and R. A. Calogero, “Microarray data analysis and mining approaches,” *Briefings Funct. Genomics Proteomics*, vol. 6, no. 4, pp. 265–281, 2007, doi: 10.1093/bfgp/elm034.

[17] B. Efron, R. Tibshirani, J. D. Storey, and V. Tusher, “Empirical bayes analysis of a microarray experiment,” *J. Am. Stat. Assoc.*, vol. 96, no. 456, pp. 1151–1160, 2001, doi: 10.1198/016214501753382129.

[18] C. P. Kolbert, W. R. Taylor, K. L. Krajnik, and D. J. O’Kane, “Gene expression microarrays.,” *Methods Mol. Med.*, vol. 85, pp. 239–255, 2003, doi: 10.1385/1-59259-380-1:239.

[19] S. Saha, S. Bandopadhyay, A. Ghosh, and K. N. Dey, “An improved fuzzy based approach to impute missing values in DNA microarray gene expression data with collaborative filtering,” *2016 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2016*, pp. 911–916, 2016, doi: 10.1109/ICACCI.2016.7732161.

[20] R. E. Hayward, J. L. DeRisi, S. Alfadhli, D. C. Kaslow, P. O. Brown, and P. K. Rathod, “Shotgun DNA microarrays and stage-specific gene expression in Plasmodium falciparum malaria,” *Mol. Microbiol.*, vol. 35, no. 1, pp. 6–14, 2000, doi: 10.1046/j.1365-2958.2000.01730.x.

[21] A. Sánchez and M. C. R. De Villa, “A Tutorial Review of Microarray Data Analysis,” *Bioinformatics*, pp. 1–55, 2008.

[22] M. Greener, “Defeating malaria: new weapons against one of our deadliest foes,” *Prescriber*, vol. 31, no. 7–8, pp. 18–22, 2020, doi: 10.1002/psb.1856.

[23] “Anopheles - Wikipedia.” https://en.wikipedia.org/wiki/Anopheles (accessed Dec. 25, 2021).

[24] Y. Yang, S. J. Adelstein, and A. I. Kassis, “Target discovery from data mining approaches,” *Drug Discov. Today*, vol. 17, no. October 2017, pp. S16--S23, 2011, doi: 10.1016/j.drudis.2011.12.006.

[25] N. Dholakia, P. Dhandhukia, and N. Roy, “Screening of potential targets in Plasmodium falciparum using stage-specific metabolic network analysis,” *Mol.*

Divers., vol. 19, no. 4, pp. 991–1002, 2015, doi: 10.1007/s11030-015-9632-0.

[26] M. Eskandari *et al.*, "Identifying the Most Appropriate Pattern for Identification of Gene Expression Changes in Ovarian Cancer Using Microarray," *Iran. Red Crescent Med. J.*, vol. 21, no. 7, 2019, doi: 10.5812/ircmj.85420.

[27] K. Thearling, *Data Mining for CRM*, vol. 10. 2009.

[28] M. J. Gardner *et al.*, "Genome sequence of the human malaria parasite Plasmodium falciparum," *Nature*, vol. 419, no. 6906, pp. 498–511, 2002, doi: 10.1038/nature01097.

[29] K. G. Le Roch *et al.*, "Discovery of gene function by expression profiling of the malaria parasite life cycle," *Science (80-.).*, vol. 301, no. 5639, pp. 1503–1508, 2003, doi: 10.1126/science.1087025.

[30] M. Walker, "Drug Target Discovery by Gene Expression Analysis Cell Cycle Genes," *Curr. Cancer Drug Targets*, vol. 1, no. 1, pp. 73–83, 2005, doi: 10.2174/1568009013334241.

[31] V. S. Chauhan, C. E. Chitnis, and D. Gaur, "Introduction: An overview of malaria and Plasmodium," *Recent Adv. Malar.*, no. 1889, pp. 1–7, 2016, doi: 10.1002/9781118493816.ch1.

[32] F. Ay, E. M. Bunnik, N. Varoquaux, J. P. Vert, W. S. Noble, and K. G. Le Roch, "Multiple dimensions of epigenetic gene regulation in the malaria parasite Plasmodium falciparum: Gene regulation via histone modifications, nucleosome positioning and nuclear architecture in P. falciparum," *BioEssays*, vol. 37, no. 2, pp. 182–194, 2015, doi: 10.1002/bies.201400145.

[33] D. A. Baker, "Molecular & Biochemical Parasitology Malaria gametocytogenesis," *Mol. Biochem. Parasitol.*, vol. 172, no. 2, pp. 57–65, 2010, doi: 10.1016/j.molbiopara.2010.03.019.

[34] C. J. Canfield, "Recent advances in malaria research," *Rev. Sanid. Milit.*, vol. 30, no. 3, pp. 183–184, 1976.

[35] E. Hanssen, K. N. Goldie, and L. Tilley, "Ultrastructure of the asexual blood stages of plasmodium falciparum," in *Methods in Cell Biology*, vol. 96, no. C, Elsevier Inc., 2010, pp. 93–116.

[36] A. H. Lee, L. S. Symington, and D. A. Fidock, "DNA Repair Mechanisms and Their Biological Roles in the Malaria Parasite Plasmodium falciparum," *Microbiol. Mol. Biol. Rev.*, vol. 78, no. 3, pp. 469–486, 2014, doi: 10.1128/mmbr.00059-13.

[37] M. Yuda, S. Iwanaga, I. Kaneko, and T. Kato, "Global transcriptional repression:

An initial and essential step for Plasmodium sexual development," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 112, no. 41, pp. 12824–12829, 2015, doi: 10.1073/pnas.1504389112.

[38] D. J. Carucci, "Genomic tools for gene and protein discovery in malaria: Toward new vaccines," *Vaccine*, vol. 19, no. 17–19, pp. 2315–2318, 2001, doi: 10.1016/S0264-410X(00)00551-X.

[39] L. Hoopes, "Genetic Diagnosis: DNA Microarrays and Cancer." https://www.nature.com/scitable/topicpage/genetic-diagnosis-dna-microarrays-and-cancer-1017/ (accessed Sep. 19, 2020).

[40] M. M. Babu, "An introduction to microarray data analysis and visualization," in *Methods in Enzymology*, vol. 470, no. C, 2010, pp. 19–50.

[41] A. Schulze and J. Downward, "Navigating gene expression using microarrays - A technology review," *Nat. Cell Biol.*, vol. 3, no. 8, 2001, doi: 10.1038/35087138.

[42] H. El-shishiny and I. Emam, "Mining Drug Targets : The Challenges and a Proposed Framework," 2006.

[43] C. Shen and Z. P. Liu, "Identifying module biomarkers of hepatocellular carcinoma from gene expression data," in *Proceedings - 2017 Chinese Automation Congress, CAC 2017*, 2017, vol. 2017-Janua, pp. 5404–5407, doi: 10.1109/CAC.2017.8243741.

[44] J. H. Kim, *Personal Genome Data Analysis*. 2019.

[45] "What is Machine Learning? A definition - Expert System." https://expertsystem.com/machine-learning-definition/ (accessed Nov. 04, 2020).

[46] B. Lance, *Machine Learning with R: Expert techniques for predictive modeling, 3rd Edition*, vol. 34, no. 4. 2020.

[47] "Microarray Data Analysis Using Machine Learning Methods," *Biosystems Engineering*, 2009. https://www.neuraldesigner.com/solutions/microarray-analysis (accessed Sep. 30, 2020).

[48] S. Jauhari and S. A. M. Rizvi, "Effective Linear Discriminant Analysis for High Dimensional, Low Sample Size Data," *Ieee/Acm Trans. Comput. Biol. Bioinforma.*, vol. 11, no. 3, pp. 533–547, 2014, doi: 10.1109/TCBB.2014.2312002.

[49] E. Alpaydın, *Introduction to Machine Learning*. 2014.

[50] P. K. Mallapragada, S. Member, and R. Jin, "SemiBoost : Boosting for Semi-Supervised Learning," vol. 31, no. 11, pp. 2000–2014, 2009.

[51] L. Cunhe and W. Chenggang, "A New Semi-Supervised Support Vector Machine

Learning Algorithm Based on," pp. 638–641, 2010.

[52] C. Ben Mamoun *et al.*, "Co-ordinated programme of gene expression during asexual intraerythrocytic development of the human malaria parasite Plasmodium falciparum revealed by microarray analysis," vol. 39, pp. 26–36, 2001.

[53] B. F. C. Kafsack, H. J. Painter, and M. Llinás, "New Agilent platform DNA microarrays for transcriptome analysis of Plasmodium falciparum and Plasmodium berghei for the malaria research community," *Malar. J.*, vol. 11, pp. 1–9, 2012, doi: 10.1186/1475-2875-11-187.

[54] S. M. and M. S. L. Subarna Sinha1, Merrill Knapp1, John Pywtorak1, Greg McCain1, Kenneth Wingerden1, Colin VanDervoort1, J. Mark Gondek1,‡, Peter Madrid1, Toufan Parman1, Stephen Gerrard2,§, Jill E. Long3, Diana L. Blithe3, "Contraceptive and Infertility Target DataBase: a contraceptive drug development tool for targeting and analysis of human reproductive specific tissues†," *Biol. Reprod.*, vol. 105, no. 6, pp. 1366–1374, 2021, doi: 10.1093/biolre/ioab172.

[55] P. D'Haeseleer, D. Patrik, and P. D'Haeseleer, "How does gene expression clustering work ?," *Nat. Biotechnol.*, vol. 23, no. 12, pp. 1499–1501, 2005, doi: 10.1038/nbt1205-1499.

[56] N. N. Mohammed and A. M. AbdulAzeez, "Evaluation of Partitioning Around Medoids Algorithm with Various Distances on Microarray Data," *IEEE*, pp. 1011–1016, 2017, doi: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.155.

[57] A. S. Shirkhorshidi, S. Aghabozorgi, and T. Y. Wah, "A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data," pp. 1–20, 2015, doi: 10.1371/journal.pone.0144059.

[58] V. P. Gazi and E. Kayiş, "Comparing clustering techniques for real microarray data," *Proc. 2012 IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Mining, ASONAM 2012*, pp. 788–791, 2012, doi: 10.1109/ASONAM.2012.143.

[59] D. K. & M. R. Zahra Nazari, Asharif, and Y. S. & S. Ogawa, "A New Hierarchical Clustering Algorithm," *IEEE*, pp. 175–180, 2015, doi: 10.1007/978-3-642-29350-4_21.

[60] D. Datta, A. Konar, and R. Janarthanan, "Extraction of interaction information among genes from gene expression time series data," *2009 World Congr. Nat. Biol. Inspired Comput. NABIC 2009 - Proc.*, pp. 98–103, 2009, doi: 10.1109/NABIC.2009.5393607.

[61] M. Rubiolo, D. H. Milone, and G. Stegmayer, "Mining Gene Regulatory Networks

by Neural Modeling of Expression Time-Series," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 12, no. 6, pp. 1365–1373, 2015, doi: 10.1109/TCBB.2015.2420551.

[62] P. C. H. Ma and K. C. C. Chan, "Inferring gene regulatory networks from expression data by discovering fuzzy dependency relationships," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 2, pp. 455–465, 2008, doi: 10.1109/TFUZZ.2007.894969.

[63] M. Middendorf, A. Kundaje, C. Wiggins, Y. Freund, and C. Leslie, "Predicting genetic regulatory response using classification," *Bioinformatics*, vol. 20, no. SUPPL. 1, 2004, doi: 10.1093/bioinformatics/bth923.

[64] P. C. H. Ma and K. C. C. Chan, "A fuzzy data mining technique for the reconstruction of gene regulatory networks from time series expression data," *Proc. 2006 IEEE Symp. Comput. Intell. Bioinforma. Comput. Biol. CIBCB'06*, pp. 124–131, 2006, doi: 10.1109/CIBCB.2006.330981.

[65] R. I. Hamed, "Inferring gene interactions from microarray gene expression data using fuzzy Petri net," *2010 IEEE Int. Conf. Commun. Control Comput. Technol. ICCCCT 2010*, pp. 845–851, 2010, doi: 10.1109/ICCCCT.2010.5670723.

[66] B. Sen Chen and C. W. Li, "Analysing microarray data in drug discovery using systems biology," *Expert Opin. Drug Discov.*, vol. 2, no. 5, pp. 755–768, 2007, doi: 10.1517/17460441.2.5.755.

[67] Z. Duren, X. Chen, R. Jiang, Y. Wang, and W. H. Wong, "Modeling gene regulation from paired expression and chromatin accessibility data," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 114, no. 25, pp. E4914–E4923, 2017, doi: 10.1073/pnas.1704553114.

[68] A. Bustamam, S. Formalidin, and T. Siswantining, "Clustering and analyzing microarray data of lymphoma using singular value decomposition (SVD) and hybrid clustering," *AIP Conf. Proc.*, vol. 2023, no. 1, p. 020220, Oct. 2018, doi: 10.1063/1.5064217.

[69] M. O. Arowolo, M. O. Adebiyi, A. A. Adebiyi, and C. Aremu, "An Efficient PCA Ensemble Learning Approach for Prediction of RNA-Seq Malaria Vector Gene Expression Data Classification," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 2, pp. 1561–1569, 2020, doi: 10.11591/ijece.v11i2.pp1561-1569.

[70] P. Ramesh, S. Veerappapillai, and R. Karuppasamy, "Gene expression profiling of corona virus microarray datasets to identify crucial targets in COVID-19 patients," *Gene Reports*, vol. 22, p. 100980, Mar. 2021, doi: 10.1016/J.GENREP.2020.100980.

[71] D. C. B. Mariano, C. Leite, L. H. S. Santos, R. E. O. Rocha, and R. C. de Melo-Minardi, “A guide to performing systematic literature reviews in bioinformatics,” *GoogleScholar*, p. 63, 2017.

[72] B. Kitchenham, “Procedures for Performing Systematic Reviews,” 2004. doi: 10.5144/0256-4947.2017.79.

[73] “GEO Accession viewer.” https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE18780 (accessed Dec. 29, 2021).

[74] Draw.io, “draw.io – Diagrams for Confluence and Jira - draw.io,” 2021. https://drawio-app.com/ (accessed Dec. 29, 2021).

[75] M. E. Ritchie *et al.*, “Limma powers differential expression analyses for RNA-sequencing and microarray studies,” *Nucleic Acids Res.*, vol. 43, no. 7, p. e47, 2015, doi: 10.1093/nar/gkv007.

[76] D. P. Berrar, W. Dubitzky, M. Granzow, and S. Downes, “Introduction to Microarray Data Analysis. Chapter 1,” in *A Practical Approach to Microarray Data Analysis*, G. M. P.Berrar, Daniel , Dubitzky Werner, Ed. Springer US, 2003, pp. XVI, 368.

[77] J. Sun, K. Passi, and C. K. Jain, “Improved microarray data analysis using feature selection methods with machine learning methods,” *Proc. - 2016 IEEE Int. Conf. Bioinforma. Biomed. BIBM 2016*, pp. 1527–1534, 2016, doi: 10.1109/BIBM.2016.7822748.

[78] H. W. Ressom, D. Lakshman, S. J. Yun, S. K. Pramanik, and B. G. de los Reyes, “Microarray Data Analysis Using Machine Learning Methods,” *Biosyst. Eng.*, vol. 1, no. 1, pp. 1–32, 2009.

[79] J. Fridlyand, *Microarray data analysis*. 2012.

[80] D. Otasek, J. H. Morris, J. Bouças, A. R. Pico, and B. Demchak, “Cytoscape Automation: Empowering workflow-based network analysis,” *Genome Biol.*, vol. 20, no. 1, Sep. 2019, doi: 10.1186/s13059-019-1758-4.

Appendices

Appendix A: R Source code

```
PFGSE18780_fit2 <- eBayes(PFGSE18780_fit2, 0.01)
GSE18780Top_Table <- topTable(PFGSE18780_fit2,
                              adjust="fdr", sort.by="B", number=2500)

GSE18780Top_TableSub <- subset(GSE18780Top_Table,
                               select=c("Gene.symbol", "adj.P.Val",
                                        "MidgutControl.MidgutInfected",
                                        "MidgutInfected.SalivaryGlandControl",
                                        "SalivaryGlandControl.SalivaryGlandInfected",
                                        "SalivaryGlandInfected.MidgutControl",
                                        "AveExpr", "P.Value","F", "adj.P.Val"))
```

Figure A. 1 Source code to find top significant genes

```
DEgdist <- dist(GSE18780DEGT, method = "euclidean")
DEGHclust <- hclust(DEgdist, method = "complete")
plot(DEGHclust, main = "Cluster Dendrogram of GSE18780 Microarray Data")
bn <- GSE18780DEGT[1:40,]
bndist <- dist(bn, method = "euclidean")
bnclust <- hclust(bndist, method = "complete")
plot(bnclust, main = "Cluster Dendrogram of GSE18780 Microarray Data")
#Clustering using pearson coreletion
PearsonDist <- function(x) as.dist(1 - abs(cor(t(x))))
heatmap(GSE18780DEGT, distfun = PearsonDist,  margins = c(6,0))
#Clustering using canberra
distCan <- function(x) dist(x, method = "canberra")
heatmap(GSE18780DEGT, distfun = distCan,  margins = c(6,0))
#Distance Maximum
distMax <- function(x) dist(x, method = "maximum")
heatmap(GSE18780DEGT, distfun = distMax,  margins = c(6,0))
#Distance Using minkowski
distMink <- function(x) dist(x, method = "minkowski")
heatmap(GSE18780DEGT, distfun = distMink, margins = c(6,0))
#Distance Using Maximum Information Coefficient
dMIC <- function(x) as.dist(mine(t(x))$MIC)
heatmap(GSE18780DEGT, distfun = dMIC,  margins = c(6,0))
#Heat map using every 10th element
heatmap(GSE18780DEGT[seq(1, nrow(GSE18780DEGT), by = 10), ],  margins = c(6,0))

PFhc <- hclust(dMax(GSE18780DEGT))
plot(PFhc)
#Draw rectangles at different cut-levels,to get the desired number of clusters
plot(DEGHclust, main = "Cluster Dendrogram of GSE18780 Microarray Data")
rect.hclust(DEGHclust, k = 2)
rect.hclust(DEGHclust, k = 4)
rect.hclust(DEGHclust, k = 8)
rect.hclust(DEGHclust, k = 10)

```

Figure A. 2 Hierarchical Clustering with Proximity Measure Source Code

Appendix B: Different Plots

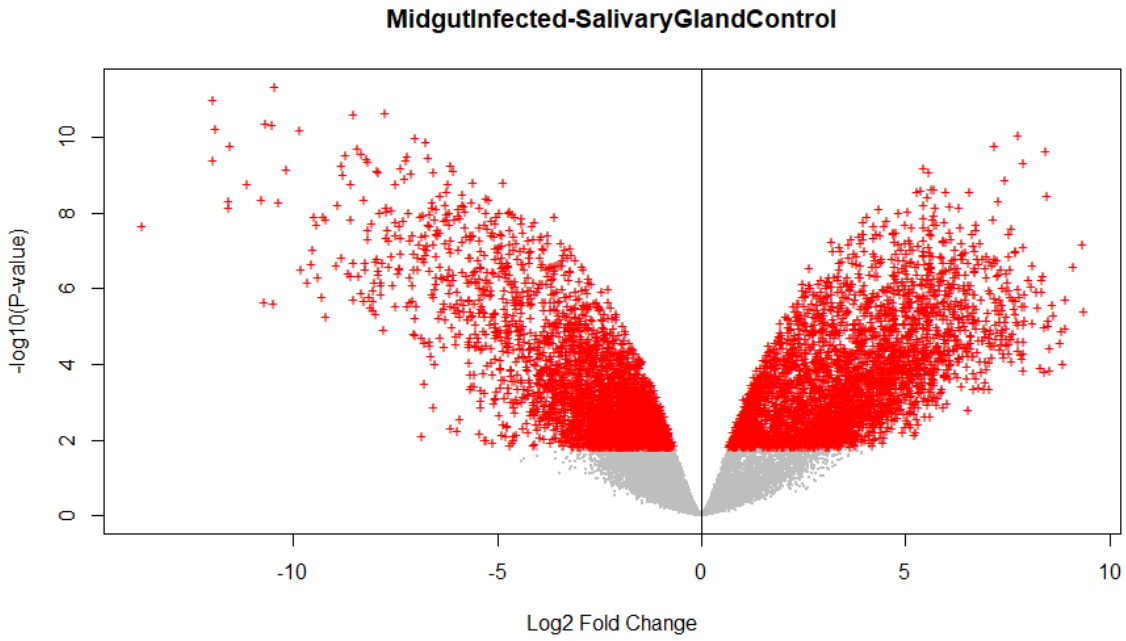


Figure A. 3 Volcano Plot

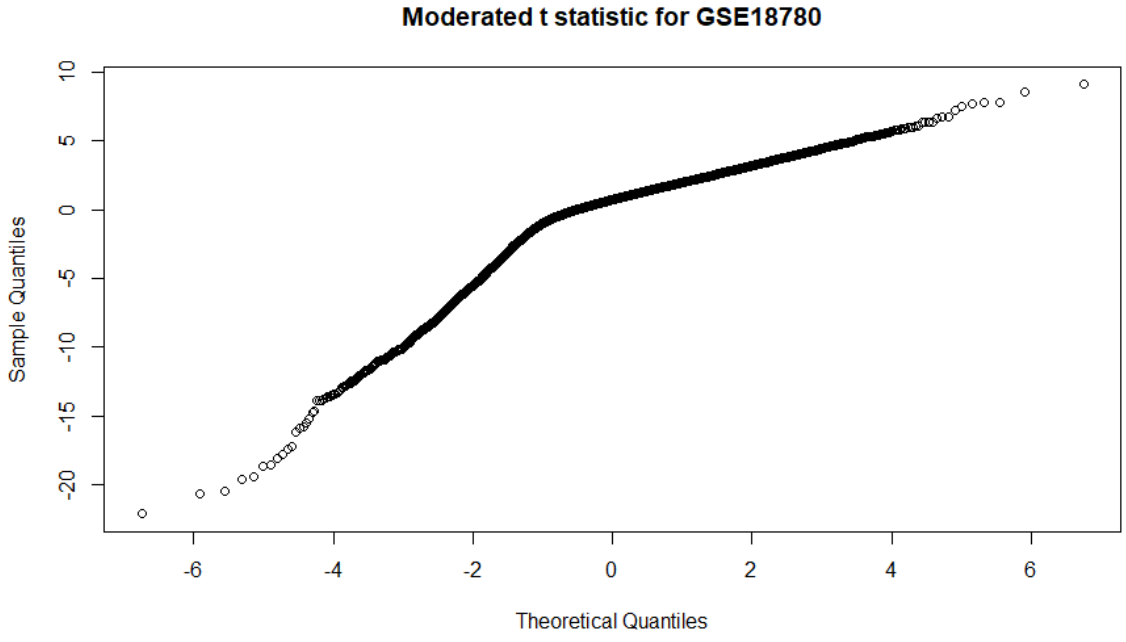


Figure A. 4 Q-Q plot

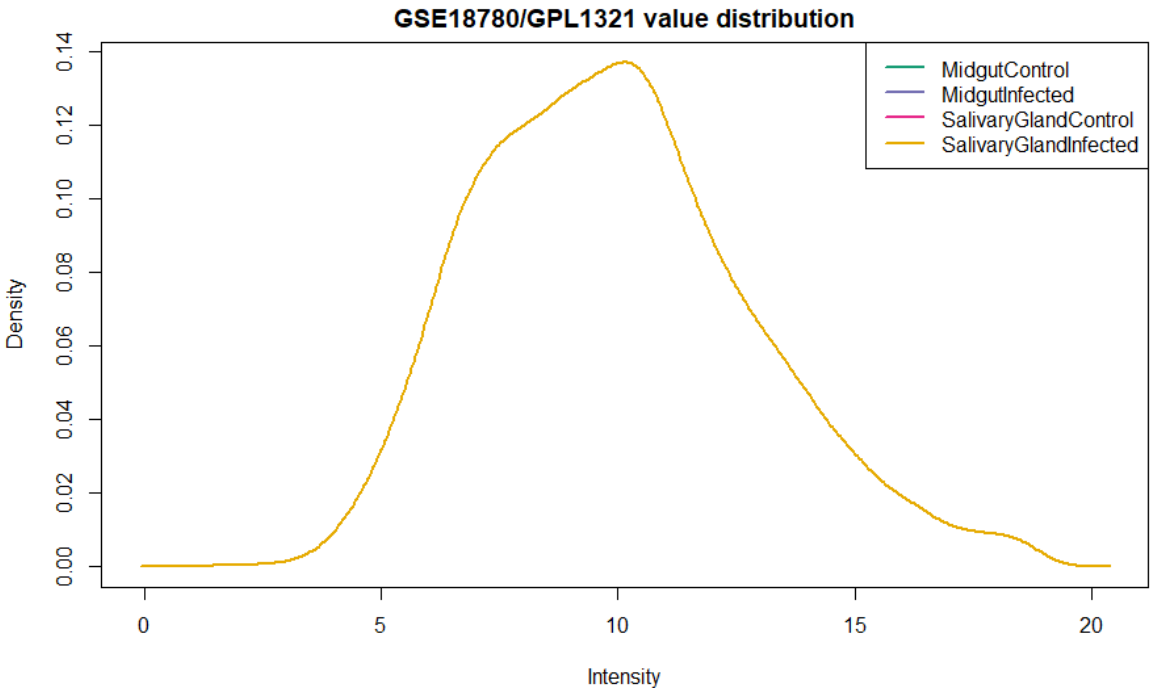


Figure A. 5 Density Plot