A MACHINE LEARNING MODEL FOR PREDICTING ESSENTIAL GENES FROM *PLASMODIUM FALCIPARUM* METABOLIC NETWORK

STEPHEN NPOANDAN BINAANSIM
(21PBF02260)
Bachelor of Science (B. Sc.) Mathematics,
Kwame Nkrumah University of Science and Technology (KNUST)

AUGUST, 2023

A MACHINE LEARNING MODEL FOR PREDICTING ESSENTIAL GENES FROM *PLASMODIUM FALCIPARUM* METABOLIC NETWORK

BY

STEPHEN NPOANDAN BINAANSIM
(21PBF02260)
Bachelor of Science (B. Sc.) Mathematics,
Kwame Nkrumah University of Science and Technology (KNUST)

A DISSERTATION SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES IN PARTIAL FULFILLMENT OF THE REQUIREMENT OF MASTER OF SCIENCE (M.Sc.) DEGREE IN BIOINFORMATICS IN THE DEPARTMENT OF COMPUTER AND INFORMATION AND INFORMATION SCIENCES, COVENANT UNIVERSITY, OTA, OGUN STATE, NIGERIA

AUGUST, 2023

ACCEPTANCE

This is to attest that this dissertation is accepted in partial fulfilment of the requirements for the award of the degree of Master of Science in Bioinformatics in the Department of Computer and Information Science, College of Science and Technology, Covenant University, Ota, Ogun State, Nigeria.

Miss Adefunke F. Oyinloye **Signature and Date**
(Secretary, School of Postgraduate Studies)

Prof. Akan B. Williams **Signature and Date**
(Dean, School of Postgraduate Studies)

DECLARATION

I, **BINAANSIM, STEPHEN NPOANDAN (21PBF02260)** declare that this dissertation is a representation of my work and implemented by me under the supervision of Professor Jelili O. Oyelade of the Department of Computer and Information Sciences, Covenant University, Ota, Ogun State, Nigeria. I attest that this dissertation has not been submitted either wholly or partially to any other university or institution of higher learning for the award of master's degree. All sources of the data and scholarly information from published and unpublished literature used in this dissertation has been duly acknowledge.

STEPHEN NPOANDAN BINAANSIM

Signature and Date

CERTIFICATION

This is to certify that this dissertation titled "**A MACHINE LEARNING MODEL FOR PREDICTING ESSENTIAL GENES FROM *PLASMODIUM FALCIPARUM* METABOLIC NETWORK**" is original research carried out by **STEPHEN NPOANDAN BINAANSIM, (21PBF02260)** and meets the requirement and regulations governing the award of Master of Science (M. Sc.) degree in Bioinformatics from the Department of Computer and Information Sciences, Covenant University, Ota, Ogun State, Nigeria under the supervision of Professor Jelili O. Oyelade, and its approved for its contribution to knowledge and literary presentation.

Prof. Jelili O. Oyelade
(Supervisor) **Signature and Date**

Prof. Olufunke O. Oladipupo
(Head of Department) **Signature and Date**

Prof. Akinwale, Adio Taofiki
(External Examiner) **Signature and Date**

Prof. Akan B. Williams
(Dean, School of Postgraduate Studies) **Signature and Date**

DEDICATION

I dedicate this project to God almighty for the supply of wisdom, strength and guidance all through this master's degree program and the carrying out this research project. Furthermore, to all Biomedical Informaticians who have paved the way to insightful research investigations in this direction.

ACKNOWLEDGEMENTS

I extend my eternal gratitude to the unwavering and faithful presence of God, who has guided me to triumph through various life stages and granted me the capacity to pursue this program and undertake this project. I offer sincere thanks to Dr. David O. Oyedepo, the Chancellor of Covenant University, whose life and ministry have profoundly impacted me during my time at the university. Your embodiment of principles such as leadership, hard work, positive mindset, and strong work ethics has been an invaluable lesson.

I am equally appreciative of the efforts of the university's Vice Chancellor and their dedication to advancing this esteemed institution. My heartfelt acknowledgments extend to Prof. Akan B. Williams, Dean of the School of Postgraduate Studies, and Prof. Olufunke O. Oladipupo, Head of the Department of Computer and Information Science, for their tireless contributions to both students and the entire academic community.

My gratitude knows no bounds for Professor Jelili O. Oyelade, whose unwavering support, guidance, encouragement, and prayers have been pivotal throughout this project. I also extend my heartfelt gratitude to the PG Coordinator of the Computer and Information Science department, whose dedication and work ethics served as a motivating example. Your dedication to shaping Bioinformatics into a transformative experience is commendable and inspiring.

I hold in high regard the Covenant University Postgraduate Student Fellowship (CUPGF), where I had the privilege of serving as both Student Chaplain and General Secretary. To my fellow executives, who became a close-knit family, I am deeply thankful. My appreciation goes out to Triumphant Dunbar, Faith Adegoke, Sharon Abimbola, Adebesin Joshua, Flourish Folorunsho (My Secretary and able assistant), Favor Folorusho, Ayoola David, Emmanuel Ogunkolati, Ojukwu Chukwuebuka Noble, Damilola Adaniji, Peace Durojaye, Murewa Newo, and Derah Mercy (Chaplain Female).

I wish to recognize my friends and course mates, whose unwavering support has been a consistent wellspring of motivation. To Jumoke Adeyemi, Samuel Kwesi Owusu-Ansah Terwiah, Faith Adegoke, Paul Owolabi, and Erika Baiguerel, among others, I express my heartfelt gratitude. To my friends and cherished associates, including David Agbeko, Isaac

Ofori, Awoonor Elijah, and numerous others whose names space constrains me from mentioning, I deeply appreciate your presence in my journey.

Lastly, I am profoundly thankful to my biological parents, Bishop Simon N. Binaansim and the late Mrs. Joyce Yawa Binaansim, who have steadfastly supported me throughout my academic pursuits. To my brother Emmanuel Binaansim and Esther Muyen Nboraki, whose friendship continues to uplift me, I extend my sincere thanks for your unwavering support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AUROC	Area Under the Receiver Operating Characteristic curve
BC	Betweenness Centrality
BiGG	Biochemical Genetic and Genomic database
CB	Constraint-Based
CC	Closeness Centrality
CDC	Centers for Disease Control
COBRA	Constraint-based modeling package
COBRApy	COBRA for Python
CoC	Clustering Coefficient
CRISPR	Clustered Regularly Interspaced Short Palindromic Repeats
CSV	Comma Separated Value
DC	Decision Tree
DC	Degree centrality
DL	deep learning
FBA	Flux Balance Analysis
FWRC	Flux-Weighted Reaction-Centric
GEM	Genome-scale Modeling
GPR	Gene-protein-reaction
GSMM	Genome-Scale Metabolic Model
GSMN	Genome-Scale Metabolic Network
kNN	k-Nearest Neighbourhood
LG	Logistic Regression
LP	Linear Programming
MFG	Mass Flow Graphs
ML	Machine Learning
MLP	Multilayer Perceptron
NB	Naive Bayes
OGEE	Online GEne Essentiality database
ReFeX	Recursive Feature Extraction Algorithm
RF	Random Forest
RolX	Role eXtraction
SBML	Systems biology mark-up language
SVM	Support Vector Machine
UML	Unified Modeling Language
WHO	World Health Organization

ABSTRACT

The accurate prediction of essential metabolic genes (i.e., genes necessary for cell survival) in eukaryotic organisms is still a difficult task in bioinformatics, especially in pathogenic species like *Plasmodium falciparum*, the malaria causing parasite. The difficulty and cost in time and resources of experimental methods has necessitated the use of computational methods for gene essentiality prediction. The majority of earlier research in this field concentrated on prokaryotes, omitting the complexity of weighted and directed metabolite transport in metabolic networks. To overcome this limitation, we developed a Network-based Machine Learning framework that explored various network properties in *Plasmodium falciparum* using the Genome-Scale Metabolic Model (iAM_Pf480) adopted from the BiGG database and essentiality data from the Ogee database. Our machine learning framework significantly increased the accuracy of gene essentiality predictions by taking into account the weighted and directed nature of the metabolic network and utilising network-based features, producing state-of-the-art results with an accuracy of 0.85 and AuROC of 0.7. This study expanded our knowledge of the complex nature of metabolic networks and their critical function in determining the essentiality of genes. Notably, our model identified important genes that were previously classified as non-essential in the Ogee database but predicted to be essential. Some of these genes have previously been linked to potential drug targets for the treatment of malaria, providing promising new research directions.

Keywords: Gene essentiality, Constraint-based Analysis, Graph-based analysis, Machine Learning, Metabolic Networks

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

The basic function of every biological system is metabolism, which produces energy, the components required for cellular development and adaptation, and controls a number of biological activities (Sahu *et al.*, 2021). Cellular metabolism is a collection of enzymatic events connected to broad functional reactions that occur inside a living cell (Rigoulet *et al., 2*020). As several anabolic and catabolic based mechanisms are of great significance to the growth of the cell and its survival, metabolism is proven to be a key factor in the treatment of various infectious diseases (Abdel-Haleem *et al., 2*018). Metabolomics has gained popularity in recent years as the field that offers the most comprehensive understanding of physiological processes happening within living cells. In a range of sectors, from clinical to environmental or even agricultural sciences, it offers useful information for disease diagnosis, toxicological research, and therapy follow-up or optimization (Cuperlovic-Culf, 2018).

Metabolic networks are a type of biological networks in which a large number of concurrent chemical reactions and transport activities connect chemical molecules and other small chemical species, known as metabolites (Rigoulet *et al., 2*020). The study of metabolic modeling, also known as metabolic pathway analysis or metabolic network analysis, has made it possible to replicate various intracellular and intercellular processes to better understand how organisms work at the systemic level. The properties of the metabolic network at the structural, kinetic, and regulatory levels may be inferred from measurements of metabolite concentrations and reaction fluxes (Ferreira, Sousa Silva, and Cordeiro, 2021). These networks are a part of the body of knowledge referred to as network medicine which plays a significant role in the medical sciences. These networks have been reconFigured in research in order to determine which pharmacological therapy-induced changes in network topography can be harmful to the pathogen (Shen *et al., 2*010).

A genome-scale metabolic network/s (GSMN/s) is a model constructed to acquire an understanding of the metabolic network. They are mathematical representations of metabolic networks that are developed from the context-specific annotated genome of a cell/organism (Chiappino-Pepe, Pandey, and Billker, 2021). A list of all biochemical processes/reactions in the cell is put together with information on cellular borders, a biomass reaction, and exchange reactions with the organism's/cell's environment in order to rebuild GSMNs, either manually or semi-manually (Freischem, Barahona, and Oyarzún 2022; Iranzadeh and Mulder, 2019). The mathematical model comprises constraints as reactions' upper and lower bound, making it more context specific which is one of the fundamental limitations for GSMNs that we will later look at (Chiappino-Pepe *et al., 2*021; Schinn *et al., 2*021). To define realistic metabolic behavior, they limit the availability/flux of nutrients and other metabolites that flow through a reaction per time which gives insight on the state of the organism (Hameri *et al., 2*019). In order to identify and forecast potential metabolic fluxes in GSMN, Flux Balance Analysis (FBA), a constraint-based (CB) modeling technique, is one of the mathematical optimization techniques used in the study of metabolic models (Wu *et al.*, 2016). This has been found among other applications in gene essentiality studies. Because of its breadth and applicability, CB Modeling of metabolism is expanding dramatically, and its integration with omics data offers mechanistic insights into the genotype-phenotype environment relationship (Bordbar *et al., 2*014).

The word "essential genes" refers to genes that are necessary for a cell to survive. For example, figuring out a gene's function in a network utilizing systems biology, evaluating metabolic microenvironments, drug target discovery and creating microorganism strains that have been biologically altered all calls for the identification of these essential genes. The environment of a cell and function of the gene determine whether it is "vital for survival" in a cell or not (Aromolaran *et al., 2*020; Nandi, Subramanian, and Sarkar, 2017). Experimental methods like transposon mutagenesis, single gene deletion, antisense RNA, and Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR) are the methods often used to determine essential genes. However, these experimental approaches tend to be more expensive, intense and time-consuming. Hence, computational techniques have been proposed to predict essentiality which is cost effective and can serve as a preliminary step to kick-start biological research into gene essentiality. In recent years, this method has become very popular (Beder *et al., 2*021; Nandi, Ganguli, and Sarkar, 2020; Nandi *et al., 2*017).

While the application of FBA in gene essentiality studies in prokaryotes has produced encouraging results and led to more advanced gene essentiality research, FBA's ability in pathogenic eukaryotes is substantially more limited (Gatto *et al., 2*015). This is partly due to the limited quality of the available Genome-scale metabolic models (GSMM/s) of eukaryotes that serves as impute to FBA and the fact that the prediction accuracy of FBA is quite sensitive to the biomass (i.e. objective function) that needs to be constantly adjusted to fit the environmental conditions under consideration. Growth rate maximization is typically the biomass function (That's we assume that the cell will do all it can to maximize growth in any environmental condition). Although additional objective functions, such the maximizing of ATP production and the lowering of substrate absorption rate, have also been proposed, It is still unclear if this set objective works effectively across different species and/or in different environmental conditions (Dusad *et al., 2*021). It is also unknown if deletion strains continue to try to optimize growth or whether gene deletions change cell physiology to achieve alternative survival aims that is not currently known (Nandi *et al., 2*020).

Lately, there has been an increase in awareness of the considerable promise that integrating FBA with machine learning provides for removing some of the core limitations of GSMN models and the traditional FBA (Ferreira *et al., 2*021; Machicao *et al., 2*021; Sahu *et al., 2*021; Zampieri *et al., 2*019). Machine learning (ML), a statistical techniques that let computers "learn" internal systems from training data and produce incredibly accurate predictions or classifications, have started to be used in GSMN research in recent years (Cheng *et al., 2*013; Wu *et al., 2*016). Numerous research studies and surveys have been done to determine if ML and DL techniques may be used in metabolic network research (Beder *et al., 2*021; Vijayakumar, Rahman, and Angione, 2020; Yu *et al., 2*017).

Graph theory has emerged as an additional approach to gain a deeper understanding of Metabolic Networks. In this method, these networks are represented as graph structures, and the features of these graphs are analyzed to provide valuable biological insights into cell metabolism. Traditionally, metabolic networks are modeled as undirected, bipartite graphs, where nodes represent both reactions and metabolites, and the graph is unweighted (Oyelade *et al., 2*018, 2019; Plaimas, Eils, and König, 2010).

However, this modeling approach does not naturally capture the concept of flux distribution, which must of necessity include its flow and directionality which is essential for understanding the flow of metabolites in the network. To address this limitation and provide a more comprehensive representation, Mass and Network Flow graph algorithms was introduced (Beguerisse-Díaz *et al., 2*018). In this model, nodes represent reactions, and the edges between them represent the flux flow from source to target reaction nodes, thus enabling a more accurate representation of metabolic processes in the cell.

In 2018, Beguerisse-Díaz *et al.* introduced an innovative framework known as Mass Flow Graphs (MFG) to construct flux-based graphs using organism-wide metabolic networks (Beguerisse-Díaz *et al., 2*018). These graphs are designed to represent the directionality of metabolic flows, where edges indicate the flow of metabolites from source to target reactions. The methodology allows for the application of flux distributions computed through FBA with or without a specific biological context. When applied to model the metabolic network of Escherichia coli bacteria, the authors observed that the flux-dependent graphs exhibited systematic changes in their topological and community structure under different environmental conditions and genetic perturbations. These changes provided valuable insights into the re-routing of metabolic flows and the varying importance of specific reactions and pathways. Such insights are crucial for understanding essential reactions and their enzymatic catalysis.

In 2022, Freischem *et al.* adopted this approach and proposed a novel machine learning method to directly predict gene essentiality from wild-type flux distributions, without assuming optimality of deletion strains (Freischem *et al., 2*022). Their approach involved projecting the wild-type FBA solution onto a mass flow graph of E. coli bacteria and training binary classifiers on the connectivity features of graph nodes to predict gene essentiality. However, this approach has not yet been explored in pathogenic eukaryotic organisms. Additionally, the impact of other connectivity features on gene essentiality prediction has not been investigated.

Millions of instances of malaria are caused by the parasitic eukaryotic organism *Plasmodium falciparum*, which has a disproportionately negative effect on low- and middle-income African nations (Abdel-Haleem *et al., 2*018). The worrying rise in malaria infections and fatalities has been highlighted in the most recent World Malaria Report. Malaria cases were reported in 247 million cases in 2021, which is a little increase from the 245 million cases found in 2020. The anticipated number of fatalities from malaria rose from 625,000 in 2020 to 619,000 in 2021 (Centers for Disease Control (CDC), 2021; World Health Organization (WHO), 2023).

The urgent need for efficient antimalarial medicines is highlighted by the increased risk of severe malaria and death experienced by certain vulnerable populations, such as small children and pregnant women. As a result, there is a rising interest in malaria research in Africa since it is crucial to solve this health issue (Chiappino-Pepe *et al., 2*021; Oyelade *et al., 2*018). Exploring important *Plasmodium falciparum* genes is critical to solving this problem since doing so opens up the possibility of creating more effective antimalarial drugs. Understanding the importance of certain genes enables researchers to focus on crucial enzymatic activities and pathways, resulting in the creation of new and improved malaria therapies.

1.2 Statement of the Problem

Finding novel drug targets involves identifying the genes necessary for certain metabolic pathways, which has significant effects on biological research. Despite this, it is challenging to classify these important genes exactly because there are so many different factors that determine whether a gene is necessary (Beder *et al., 2*021). These variables include type of species, the phenotypes under investigation, and environmental conditions.

The more conventional experimental methods need a lot of time and money to determine which genes are essential. To get around these limitations, constraint-based computational techniques like Flux Balance Analysis (FBA) have been used. Even though FBA has been shown to be effective in finding crucial genes in prokaryotic species, its application to eukaryotic organisms has not been explored. Moreover, its success remains linked to the biomass equation while it only considers one environmental factor at a time (Aromolaran, Aromolaran, *et al., 2*021; Freischem *et al., 2*022).

Network-based machine learning methods have demonstrated their effectiveness in identifying metabolically important genes in prokaryotic organisms, which addresses the constraints of FBA. It has been demonstrated that these methods are helpful in identifying metabolically important genes. However, there are limited studies on the effectiveness of these approaches in eukaryotic organisms which this research seeks to address.

1.3 Research Questions

This research seeks to answer the following questions:

i. How do we apply network-based ML framework in gene essentiality prediction in the eukaryotic species?

ii. Which network-based properties extracted from flux-weighted reaction-centric graphs can effectively predict these metabolic essential genes?

iii. How does the developed framework compare to traditional constraint-based computational methods and current research findings in this domain in terms of its accuracy and efficiency for identifying metabolic essential genes in eukaryotic pathogens?

1.4 Aim and Objectives of the Study

The aim of this research study is to develop a network-based machine learning framework that combines the features of traditional FBA for predicting metabolically essential genes in eukaryotic *Plasmodium falciparum* pathogens.

The objectives of this study are to:

i. Create flux-weighted reaction-centric (FWRC) graphs using the genome-scale metabolic models of *Plasmodium falciparum* organisms model extracted from BiGG database.

ii. Extract predictive network-based features from the flux-weighted reaction-centric graphs that are predicted by metabolic essential genes.

iii. Train and test machine learning models using the extracted features to predict metabolic essential genes in the chosen model organisms and evaluate their performance across various datasets.

iv. Evaluate the performance of the developed framework by comparing it to traditional constraint-based computational methods in terms of accuracy, and efficiency.

1.5 Research Methodology

Considering our research objectives above, I will be employing the following methods to achieve them.

Objective 1: Create flux-weighted reaction-centric graphs using the genome-scale metabolic models of *Plasmodium falciparum* and Saccharomyces cerevisiae organisms extracted models, where reactions are represented as nodes and metabolites as edges.

a) Identify and download the iAM_Pfal480 of *Plasmodium falciparum* 3D7 constructed by Abdel-Haleem *et al.* (2018) from BiGG database (http://bigg.ucsd.edu/).
b) Construct a flux-weighted reaction-centric network (FWRC) of the GSMMs adopting the mass flow graph algorithm proposed by (Beguerisse-Díaz *et al., 2*018) using COBRApy and NetworkX python library.

Objective 2: Identify and extract predictive network-based features from the flux-weighted reaction-centric graphs that are predicted by metabolic essential genes.

a) The constructed network graphs will be exported as a network file, and a set of features extracted using the NetworkX package of the python programming language.
b) Save features in a csv file of reaction nodes and feature columns.

Objective 3: Train and test machine learning models using the extracted features to predict metabolic essential genes in the chosen model organisms and evaluate their performance across various datasets.

a) Employ various popular ML techniques such as Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Naive Bayes (NB), Logistic Regression (LR) and k-Nearest Neighbour using *Scikit-learn* package.

b) Train and test them across the metabolic network features that have being extracted.

c) Performance evaluation metrics (accuracy, recall, F1- score, precision) would be carried out on the machine learning models using the *Scikit-learn* python library taking into account the best performing feature combinations and ML model in each model organism.

Objective 4: Evaluate the performance of the developed framework by comparing it to traditional constraint-based computational methods in terms of accuracy, and efficiency.

a) Carry out a single gene/reaction deletion flux balance analysis of iAM-Pf480 using the COBRApy python tool to predict essential genes and perform a comparative analysis with our ML prediction.

1.6 Significance of the Study

The successful development of this framework will contribute to identification of essential genes that will lead to advancements in drug target discovery, particularly in the field of combating eukaryotic pathogens, and provide valuable insights into the metabolic networks of these organisms. For example, the ability to precisely identify which genes are vital in *Plasmodium falciparum* which is the parasite that claims the most lives when it comes to malaria and caused serious health issue across the world (World Health Organization (WHO) 2023) is particularly crucial for the development of novel medications and treatment strategies to combat malaria. When mass flow graphs, network science, and machine learning (ML) algorithms are used together, it will be easier and more accurate to predict essentiality of a gene. The research seeks to investigate this possibility and show how these methods can be used to study GSMMs in these organisms.

1.7 Scope of the Study

This study investigates metabolic networks in eukaryotic pathogen using a network-based machine learning technique, with a particular emphasis on metabolic essential genes prediction. The study obtains genome-scale metabolic models for the model species *Plasmodium falciparum*. These models will be used to generate flux-weighted reaction-centric network graphs, where the shared metabolites function as edges and reactions as nodes. To accurately predict gene essentiality, the study will also entail the discovery and extraction of predictive characteristics from the produced graphs.

This research will focus on the use of ML algorithms and network science techniques to analyze the GSMMs of *Plasmodium falciparum*, a unicellular parasite that causes malaria. The scope of the research is to get an understanding of which genes are essential.

1.8 Limitations of the Study

This study seeks to deploy a Network-based machine learning framework for predicting eukaryotic organisms. Our study emphasizes *Plasmodium falciparum* which may restrict the applicability and generalizability of its results to the entire eukaryotic kingdoms. Further research would be required to see whether the framework is useful in finding metabolic essential genes in other eukaryotic species that have not being researched.

1.9 Organization of Dissertation

This dissertation has been structured into five chapters, chapter one gives us the background of the study, revealing the research aims and objectives. Chapter two includes review of literature on the theoretical foundation of the research and introduces the concepts of metabolic gene essentiality prediction. It concludes with related works that have been carried out. Chapter three records the methodology. Chapter four presents the results and discuss our research findings. Chapter five concludes with recommendations for future research.

CHAPTER TWO
LITERATURE REVIEW

2.1 Preamble

This chapter reviews a theoretical review of metabolic networks, Metabolic gene essentiality prediction, flux balance analysis, and network-based approaches to gene essentiality prediction. The study also reviewed related works in this area.

2.2 Theoretical Review

This review discusses the theoretical background to the study, elaborating on metabolic network analysis and their biological relevance, gene essentiality prediction and finally the application of ML and network science techniques in essentiality prediction.

2.3 Metabolic Network Analysis

A metabolic network is a form of biological network in which a large number of concurrent chemical reactions and transport activities connect chemical molecules and other small chemical species, known as metabolites (Medlock and Papin, 2020). The study of metabolic modelling, also known as metabolic pathway analysis or metabolic network analysis, has made it possible to replicate various intracellular and intercellular processes to better understand how organisms survive. The properties of the metabolic network at the structural, kinetic, and regulatory levels can be inferred from measurements of metabolite concentrations and reaction fluxes (Freischem *et al., 2*022). These networks are a part of the body of knowledge referred to as network medicine. These networks have been reconFigured in order to determine which pharmacological therapy-induced changes in network topography are harmful to the pathogen (Shen *et al., 2*010).

2.3.1 The Importance of Metabolic Network Analysis in Biological Research

By offering important insights into the intricate and interrelated biochemical processes that take place within live organisms, metabolic network analysis plays a significant role in biological research. Several factors make metabolic network analysis crucial in biological research, including the following:

i. **Understanding Cellular Function:** The web of interrelated biochemical processes that take place inside cells is represented by metabolic networks (Rigoulet *et al., 2*020).

Researchers can better understand how cells function, including energy metabolism, the production of vital chemicals, and food utilization, by investigating these networks. Understanding cellular physiology and discovering possible treatment targets require this information (Bandoim, 2019).

ii. **Systems-Level View:** Metabolic network analysis enables researchers to approach biology from a systems-level perspective (Bernstein *et al., 2*021). By taking into account the connections and regulatory mechanisms between various metabolic processes, emergent feature research and the discovery of global metabolic phenotypes are both made possible. This all-encompassing viewpoint offers a thorough understanding of cellular metabolism and assists in revealing unrecognized connections (Buchweitz *et al., 2*020)

iii. **Modeling and Prediction:** Using computational techniques and experimental data, we build metabolic network models. With the use of these models, which replicate how metabolic pathways behave, scientists are able to predict metabolic fluxes, patterns of food/energy utilization, and the results of genetic or environmental perturbations. Such prediction abilities are crucial for comprehending metabolic dynamics and supporting attempts to develop new drugs or improve metabolic engineering research (Voit, 2017).

iv. **Disease Processes and Biomarker Discovery:** By highlighting changes in metabolic pathways linked to diverse sicknesses, metabolic network analysis aids in the understanding of disease processes. Researchers can find critical metabolites, enzymes, or pathways that are dysregulated by contrasting healthy and sick metabolic states, which can lead to the identification of novel diagnostic biomarkers or therapeutic targets (Toubiana *et al., 2*019). Metabolic network analysis has the potential to revolutionize personalized treatment. Researchers may create customized metabolic models by combining patient-specific data from genetics, metabolomics, and clinical data (Surendran *et al., 2*022). These models can forecast how a person's metabolism will react to particular medications or dietary changes, making it easier to create personalized treatment plans (Masutin, Kersch, and SchmitzSpanke, 2022).

Metabolic network analysis helps with the design and optimisation of microbial cell factories for biotechnological applications in synthetic biology and biotechnology. Researchers can improve the synthesis of desired substances, such as biofuels, medicines, or industrial chemicals, while minimizing undesirable byproducts or maximizing resource utilization by analyzing and altering metabolic pathways (Cakmak and Celik, 2021).

2.3.2 Metabolic Gene Essentiality Prediction

As efficient alternatives to these exceedingly difficult experimental methodologies for discovering essential genes, scientists are increasingly using computational methods that rely either on constraint-based techniques, homology mapping, or ML approaches (Acencio and Lemke, 2009; Azhagesan, Ravindran, and Raman, 2018; Campos *et al., 2*022; Zhang, Acencio, and Lemke, 2016). The algorithms that are homology-based and used for essential gene prediction suggest that essential genes are typically shared by distantly related species, will not likely mutate, and tend to remain unchanged (Hua *et al., 2*016; Schonfeld *et al., 2*021). Through comparative genomic research, essential genes in distinct bacterial species have been discovered (Aromolaran *et al., 2*020; Nandi *et al., 2*020). However, the fact that ortholog genes that are conserved across species make up such a small portion of the whole genome limits this method. It is also proven that genes that are majorly conserved between species are not always essential since the environmental factors that an organism is exposed to locally also have an influence on the essentiality of its genes (Nandi *et al., 2*020).

FBA and another constraint-based modelling (CBM) method compute steady-state metabolic fluxes using genome-scale reconstructed metabolic networks. This method is often used to find essential genes by doing a computer-based gene deletion (called an "in-silico knockout") and looking at how it affects the pathogen’s survival (Plata *et al., 2*010; Wu *et al., 2*016). However, this approach has limitations since only a small number of environmental conditions are considered for a given biomass (or target function) (Wu *et al., 2*016). On the other hand, machine learning (ML) techniques are a variety of data-driven approaches that create predictions for unlabelled data from a model based on the underlying patterns of training data. These machine learning methods may generally be broken down into supervised, semi-supervised, and unsupervised procedures. For model training, supervised approaches such as Naive Bayes, Decision Tree, and Support Vector Machine (SVM) require a significant quantity of labelled data (Hua *et al., 2*016). In contrast, the unsupervised methodology uses clustering methods that don't need labelled data, such as K-Means Clustering.

Semi-supervised ML systems that provide a hybrid of the benefits of supervised and unsupervised ML procedures. Thanks to these strategies, the models may be trained with only a small quantity of labelled data (Nandi *et al., 2*020). For these machine learning classifiers to make more accurate predictions, the hyperparameter has to be improved. There are several meta-heuristic methods that have been used for hyper-parameter tuning, including Genetic Algorithm (GA), Particle Swarm Optimisation (PSO), Grey Wolf Optimizer (GWO), Ant Colony Optimisation (ACO), Ant Lion Optimizer (ALO), and others (Nandi *et al., 2*020).

Based on the availability of labelled data for most genes, researchers are using supervised ML and DL algorithms to identify essential genes (Hasan and Lonardi, 2020; Hua *et al., 2*016; Schonfeld *et al., 2*021; Zampieri *et al., 2*019). These techniques have the capacity to accurately describe the hidden information and patterns of a wide variety of biologically significant "features," which is their main benefit. These "features" are diverse in nature and represent the numerous traits connected to essentiality. Several supervised machine learning classifiers have been applied to the problem of predicting whether a gene is essential. These include SVM (Cheng *et al., 2*013), ensemble methods (Yu *et al., 2*017), logistic regression (Cheng *et al., 2*013), decision tree (Cheng *et al., 2*013), random forest (Nigatu *et al., 2*017; Yu *et al., 2*017), and probabilistic Bayesian-based methods. In addition, multilayer perceptron network-based DL approaches have been used for the essential gene prediction process (Freischem *et al., 2*022). In most of these studies, the researchers have generally relied on simpler optimisation strategies, which investigate the whole space of parameters in each possible combination (Hasan and Lonardi, 2020).

These ML-based classifiers use the traits of genes that were previously annotated, verified through experimentation, and categorized as essential or non-essential to make predictions about the essentiality of unannotated genes (Nandi *et al., 2*020). To accomplish this goal, the researchers have carefully selected a variety of different feature combinations. Most ML techniques either use computed features generated from coding sequences, network topology, or both. The phyletic retention (PR), the effective number of codons (ENC), and the codon adaptation index (CAI), including the percentage of genetic content that is usually inferred from the genomic sequences, are some of the criteria that are known to exist in bacteria and that have been used to determine whether or not a gene is essential (Li *et al., 2*019; Nigatu *et al., 2*017).

To identify topological network features and classify genes according to their level of requirement, protein interaction networks, or PINs, have been used (Aromolaran, Beder, *et al., 2*021; Freischem *et al., 2*022; Kumar *et al., 2*018). These techniques, on the other hand, are useless for most animals since they contradict the centrality-lethality principle in a PIN, which hypotheses that nodes with higher centrality in a network are more likely to produce lethal phenotypes on removal compared to nodes with lower centrality (Nandi *et al., 2*020; Raman, Damaraju, and Joshi, 2013). Nevertheless, relatively few studies have classified essential genes that have been computed in a specific environmental state using flux-based features that were extracted from metabolic network graphs. This is because such classifications do not accurately represent a comprehensive set of characteristics (Freischem *et al., 2*022; Nandi *et al., 2*020, 2017). In a great number of recent pieces of research literature, in-depth analyses of the most current machine-learning techniques for evaluating the essentiality of genes have been presented (Dusad *et al., 2*021; Freischem *et al., 2*022; Zampieri *et al., 2*019).

2.3.3 Flux Balance Analysis (FBA)

FBA is one of the primary Linear Programming (LP) approaches utilized in the solution of GSMM/s Constraint-Based issues. It works by computing the best steady-state flux distribution of a cell; the flux distribution specifies the cell phenotype (Freischem *et al., 2*022; Gatto *et al., 2*015). A GSMN is used as input. The stoichiometric matrix and upper and lower constraints on reaction fluxes are used to generate a linear system of equations. In addition, the cell objective function Z is specified, which encodes the biological aim of the cell (Cuevas *et al., 2*016; Freischem *et al., 2*022). This is presumptively predicated on the idea that cell metabolism is optimized for maximum cell growth. FBA determines the solution vector to the following restricted optimization issue using linear programming:

$$Max\ Z = C^T V \tag{2.1}$$

$subject\ to$:

$$\frac{dy}{dt} = SV = 0 \tag{2.1a}$$

$$v_l \leq v \leq v_u \tag{2.1b}$$

where C encodes the cell objective function v_l and v_u are vectors containing the lower and upper limits on the fluxes of the reactions involved, respectively. Researchers are able to use FBA to determine the flux flow of the cells under different environmental and genetic conditions by altering the reaction flux bounds (Martins Conde, Sauter, and Pfau, 2016; Yasemi and Jolicoeur, 2021). FBA has been applied primarily in studies of gene essentiality prediction via single or double genes and/or reaction deletion(knock-out) through in silico simulations.

One drawback of FBA is the necessity of designing the objective function to be optimized to accurately reflect cellular physiology (Raman *et al., 2*013). Maximizing growth rate is a popular choice for any model organism in question, although it is debatable whether this is a feasible cellular target for all organisms or under all growth conditions. Other objective functions, such as maximizing ATP generation and minimizing substrate absorption rate, have been suggested, even though the great majority of FBA investigations focus on the maximization of cellular growth (Dusad *et al., 2*021; Raman *et al., 2*013). So, the question remains, "what if it is not always the case that all the cell needs is to optimize for growth under every condition?"

Secondly, amidst of all the environmental conditions that influence gene essentiality in organisms, FBA is only able to consider one experimental condition at a time and is computationally expensive especially for performing in-silico analysis; FBA gene/reaction knock-out must be performed individually for each candidate reaction, which is known as single or double-gene/reaction knockout (Freischem *et al., 2*022).

2.3.4 Network-Based Machine Learning Approaches

In order to comprehend and predict essential metabolic genes primarily through the study of various metabolic networks, there has recently been interest in exploring the interface between network science and machine learning (Dusad *et al., 2*021). Graph-theoretic notions such as degree distributions and centrality measurements may disclose structural aspects of the connectivity of the overall system, while clustering methods can expose substructures that were buried in the network topology. These types of methods may be integrated with the examination of perturbations, which can include the removal of network nodes or edges (Campos *et al., 2*022; Dusad *et al., 2*021).

These perturbations can reflect changes in the environment, gene knockouts, or medicinal medications that target particular metabolic enzymes. In contrast to FBA, which bases its analysis on the selection of a particular objective function, network approaches only rely on metabolic stoichiometry in order to conduct their calculations. In the case of complementing FBA implementation, graph modelling offers the opportunity to overcome some of its bottlenecks listed above. Unlike the FBA approach, which needs a crucial definition of an objective function that accurately defines a cell's physiological state, the network modelling approach can stimulate cell perturbation analysis directly from the stoichiometric equation (Campos *et al., 2*022; Dusad *et al., 2*021; Freischem *et al., 2*022).

2.3.5 Graph Feature Mining

Graph mining is the practise of drawing patterns, structures, and insights from graph data. Graph mining takes into consideration the innate structure of the data itself, as opposed to conventional data mining, which concentrates on organized data and frequent data values (Brahmaih 2020; Kanti Kumar, Dutta, and Kumar, 2023).

Since it enables analysis and pattern discovery inside intricate biological systems, graph mining becomes especially pertinent in the setting of biological networks (Muzio *et al.*, 2020). Protein-protein interaction networks, gene regulatory networks, and metabolic networks are examples of biological networks that may be represented as graphs, where nodes are biological entities (such genes, proteins, or metabolites), and edges are interactions or connections between them (Milano, Agapito, and Cannataro, 2022).

Graph mining algorithms are able to reveal repeating subgraphs that stand in for common motifs or functional units within the network while looking for regular chemical structures. These patterns may provide light on the structure, operation, and control mechanisms of biological systems (Takigawa and Mamitsuka, 2013).

Identifying tightly linked communities or groupings inside biological networks is another task that may be accomplished using graph mining approaches (Takigawa and Mamitsuka, 2013). To locate node clusters with robust relationships or functional correlations in biological networks, social network analysis concepts may be used (Wang, Wang, and Zheng, 2022). Understanding the modular structure of biological systems, recognising functional modules or pathways, and discovering possible biomarkers or therapeutic targets may all be aided by this approach (Camacho *et al., 2*018).

Graph mining is used to convert tabular or flat data into a property graph representation. In using graph mining methods, this translation allows the investigation and exploration of biological networks. Researchers may also use graph mining algorithms to uncover significant patterns, find related groupings, and obtain insights into the underlying biological processes by modelling biological data as graphs. Biological network graph mining provides strong tools for deciphering and analysing complicated biological data (Erciyes 2015; Muzio, O'Bray, and Borgwardt, 2020). It makes it possible to find common patterns, identify functional modules, and characterize network structures, all of which advance our knowledge of biological systems and make it easier to find new drugs, identify biomarkers, and conduct systems biology research. We examine some graph features mined from graphs that are of interest to our research into metabolic graph networks.

A. Major Network topological Features Mine from graphs

Topological network characteristics have also been accepted as sufficient justification for describing metabolic essential genes. Numerous studies have been conducted on the relationship between topological structures and biological processes (Li *et al., 2*019; Wang *et al., 2*022). The centrality-lethality hypothesis in biological networks states that particularly central nodes within a network are more likely to be critical or deadly for the general wellbeing of the biological system and have had great application in the studies of biological networks (Azhagesan *et al., 2*018). Numerous studies have examined how centrality and essentiality—a metric of a node's importance within a network—are related in biological networks (Jeong *et al., 2*001). Thus, nodes with higher centralities are more likely to be essential to the network's overall survival.

The number of a nodes near neighbours is counted in degree centrality, the most basic centrality measure. Other centrality measures, such as betweenness centrality and load centrality, evaluate a node's significance by considering its role as a conduit for traffic between different network nodes (M. Ashtiani *et al., 2*018; Jeong *et al., 2*001; Nandi *et al., 2*020; Wang *et al., 2*022).

The stability or ability of the system to survive may be dramatically impacted by the loss or malfunctioning of crucial nodes (M. Ashtiani *et al., 2*018). The performance and structure of the network are preserved by important nodes. By examining the relationship between centrality and essentiality, it is feasible to comprehend the key nodes within biological networks and their potential participation in a variety of biological processes (Christiansen, 2022).

Here we discuss some of the popular centrality features that have been applied in metabolic network studies for essentiality prediction.

i. Degree Centrality (DC):

DC is a well-known and popular feature that has been applied in gene essentiality studies. DC, which is based on the number of edges that connect a node to other nodes, is a metric for node significance in a network (Naderi Yeganeh *et al., 2*020; Wang *et al., 2*022). In other words, it represents a node's degree of network connectivity. According to the number of processes or metabolites they interact with, degree centrality may be used to determine the most crucial metabolites or enzymes in a metabolic network. Due to their numerous connections to other nodes in the network, nodes with high degrees of centrality are sometimes referred to as "hub" nodes. The given equation serves as a representation of DC.

$v_i \ as \ nodes \ of \ a \ given \ network$

Where;

$$k_i = \sum_{j \in N_i} e_i |e_{i,j}| \in E \ and \ N_i \qquad (2.2)$$

Corresponding to neighbours sets in v_i.

Due to the fact that DC can shed light on metabolic pathways and how they are regulated, it plays a crucial role in metabolic network analysis (Kim, Ashlock, and Yoon, 2019). For example, sustaining metabolic flow or controlling the whole metabolic network may depend on hub metabolites or enzymes. Changes in degree centrality may also be pointers to sickness or metabolic disorders. In order to understand the metabolic network more thoroughly, degree centrality can also be utilized in conjunction with other network metrics like betweenness centrality or closeness centrality. It is important to note that DC alone may not be a reliable predictor of network architecture since it only takes into consideration the natural surroundings of nodes (Wang *et al., 2*022).

ii. Closeness Centrality:

Closeness centrality is a measure of how quickly information can spread from one node to all other nodes in a network. In network science, it is a way of quantifying the importance of a node based on its proximity to other nodes in the network. Specifically, closeness centrality is defined as the reciprocal of the sum of the shortest path distances between a node and all other nodes in the network (M. Ashtiani *et al., 2*018).

$$C_c(v) = \frac{n-1}{\sum_{i \neq v, i \in v} d_{vi}} \tag{2.3}$$

Where:

n is the number of reaction and d_{vi} the sum of the shortest distance from v to all other nodes.

In metabolic network analysis, closeness centrality can be used to identify key metabolites that are central to the overall network. These central metabolites are likely to be essential for the proper functioning of the network, as they are involved in many metabolic pathways and can quickly influence the behaviour of other metabolites. Additionally, changes in the centrality of specific metabolites can indicate shifts in metabolic activity or the presence of regulatory mechanisms that control metabolic fluxes. Therefore, closeness centrality is a useful tool for understanding the structure and dynamics of metabolic networks and can aid in the discovery of new drug targets and metabolic engineering strategies (Wang *et al., 2*022).

iii. Betweenness Centrality (BC):

Betweenness centrality is a network science metric that measures the importance of a node in a network based on its ability to act as a bridge between other nodes. Specifically, betweenness centrality is calculated as the number of shortest paths between all pairs of nodes in a network that pass through a given node (M. Ashtiani *et al., 2*018). Nodes with high BC are considered critical for the flow of information or resources through the network.

The BC of a node i, BC (i), is mathematically represented:

$$BC(i) = \sum_{j \neq i \neq k} \frac{\sigma_{jk}}{\sigma_{ik}} \qquad (2.4)$$

Where;

σ_{jk} is the total number of shortest paths from node i to k and $\sigma_{jk}(i)$ is the total number of paths passing through node i.

In metabolic network analysis, betweenness centrality can be used to identify key metabolic pathways and enzymes that are critical for the overall functioning of the network. For example, enzymes with high betweenness centrality may be essential for connecting different metabolic pathways, regulating metabolic flux, or responding to environmental stimuli. By targeting these key nodes, researchers can potentially identify new drug targets or metabolic engineering strategies to optimize metabolic pathways in biotechnology or healthcare applications. Additionally, betweenness centrality can be used to study the robustness and resilience of metabolic networks to perturbations or mutations, which can have important implications for evolutionary and ecological dynamics (Ashtiani *et al., 2*018; Wang *et al., 2*022).

iv. Clustering Coefficient:

The clustering coefficient is a measure of the degree to which nodes in a network tend to cluster together. It quantifies the extent to which nodes in a network are connected to each other, and it is defined as the ratio of the number of connections between a node's neighbors to the maximum number of such connections that could exist. In other words, it measures how tightly connected a node's neighbors are to each other. CC is mathematically represented as:

$$CC(i) = \frac{n_i}{k_i(k_i - 1)} \qquad (2.5)$$

Where;

n_i the number of arcs or edges between neighbors of the i node, and k_i , the number of neighbors of node i.

In metabolic network analysis, the clustering coefficient is used to study the topological properties of metabolic networks. It provides information about the organization and structure of metabolic pathways and can reveal important insights into the underlying biological processes. High clustering coefficients indicate that nodes in the network tend to be highly connected, which suggests that they may be involved in similar metabolic functions or pathways. Low clustering coefficients, on the other hand, suggest that nodes are more sparsely connected and may be involved in more diverse metabolic processes. By analyzing the clustering coefficient of metabolic networks, researchers can identify key metabolic hubs, modules, and pathways that play important roles in cellular metabolism. This information can be used to predict metabolic fluxes, identify potential drug targets, and understand the overall function of metabolic networks (Naderi Yeganeh *et al., 2*020).

v. PageRank Algorithm

This algorithm, developed by Google researchers, has been used to evaluate the significance of networks and is being applied in ranking Google webpages (Naderi Yeganeh *et al., 2*020). A website's page rank is a numerical indicator of its relative relevance based on the quantity of inbound and outgoing connections. Links going to a webpage from the outside are known as inbound links (Jone, 2023). Outbound links are those that lead from one website to another. The mathematical formula for page rank algorithms is:

$$P_i = (1 - d) + d \sum_{j \in M(i)} \frac{P_i}{L(j)} \tag{2.6}$$

Where P_i is the Page rank of the webpages, $M(i)$ is the set of web pages linked to page i, $L(j)$ are the outbound linkages, and d serving as the residual probability usually set at 0.85. Pagerank algorithms are being proposed and applied in finding the importance of reaction nodes in metabolic reaction to reaction networks (Li *et al., 2*013).

B. Network Node Role Analysis

The practice of identifying and describing the various functional or structural functions that nodes (individual entities) perform inside a network is known as node role analysis (Henderson *et al., 2*011). By investigating each node's characteristics, interactions, and network placements, node role analysis seeks to identify patterns and correlations among nodes (Loem, 2021). By identifying nodes that display similar behaviours or fulfil certain tasks in the larger system, it offers insights into the organization, dynamics, and operation of the network. There are various processes involved in the study of node roles, which may change based on the particular situation and research goals:

Extracting Features: Extracting pertinent characteristics or qualities from the nodes and their connections is the initial stage. These qualities may be things like a node's degree centrality (the quantity of connections), clustering coefficient (a measure of how related a node's neighbors are), or any other pertinent metrics that describe the node's characteristics in the network. **Role Detection:** After the characteristics have been recovered, a number of techniques may be used to identify the roles present in the network. In order to categorize nodes with comparable feature profiles into separate roles or clusters, this may include using clustering algorithms, community discovery methods, or other strategies. **Role Characterization:** After roles have been identified, the next stage is to describe and interpret the roles in light of their functional or structural importance. Analyzing the patterns, traits, and actions shown by nodes within each role is necessary to do this. It could include locating key or significant nodes, nodes serving as a bridge or connection between various network segments, or nodes with particular features or behaviors. The roles that were found must be validated, and their significance must be understood in light of the network and the particular study research domain.

i. ReFeX (Recursive Feature eXtraction)

ReFeX, an algorithm proposed by (Henderson *et al., 2*011) is considered as a valuable node role analysis approach for directed graph network due to its ability to extract meaningful and transferable features from nodes in a graph. By capturing regional information and utilizing the graph's structure, ReFeX enables the identification and classification of nodes based on their characteristics, which is essential for understanding node roles within a network (Henderson *et al., 2*011, 2012).

Node role analysis aims to identify distinct functional or structural roles that nodes play within a network. It provides insights into the organization and dynamics of the network by uncovering patterns and relationships among nodes. ReFeX contributes to node role analysis by offering a systematic and effective approach to extract features that help distinguish and classify nodes, enabling the identification of their roles within the graph.

The algorithm used in ReFeX takes into account three main attributes: local, egonet, and recursive features. These attributes provide different perspectives on the nodes and their relationships, allowing for a comprehensive analysis of node roles. **Local features** capture properties directly related to the node itself, such as its degree (number of edges connected to the node) and the total degree (sum of inward and outward edges). These features reflect the node's connectivity within the network and provide insights into its importance and centrality. **Egonet features** focus on subgraphs formed by the node and its neighbouring nodes. These features consider the degree and edge count within the subgraph, providing information about the node's influence within its immediate neighbourhood. By analysing egonet features, researchers can identify nodes that act as hubs or bridges between different parts of the network.

The most significant contribution of ReFeX to node role analysis comes from recursive features. These features capture aggregated information computed over a feature value among a node's neighbours. They provide a broader perspective on the node's role by considering the characteristics and interactions of its neighbouring nodes. By generating and pruning recursive features, ReFeX identifies important patterns and dependencies within the network, contributing to the understanding of node roles. The transferability of the features extracted by ReFeX is another crucial aspect for node role analysis. The features should not only help predict properties of the given node but also be applicable to other graphs. This transferability allows researchers to compare node roles across different networks and gain insights into the general principles governing node functionality and behaviour (Henderson *et al., 2*011, 2012).

ReFeX as a node role analysis approach offers a systematic and effective method for extracting meaningful features from nodes in a directed and weighted graph. By considering local, egonet, and recursive features, ReFeX provides a comprehensive view of node characteristics, relationships, and dependencies. These features contribute to the identification and classification of node roles, enabling a deeper understanding of network organization and dynamics. The transferability of the features extracted by ReFeX allows for cross-network comparisons and generalization of node roles, providing valuable insights into node functionality and behavior (Henderson *et al.*, 2011, 2012).

ii. Role eXtraction (RolX)

Henderson *et al.*, in (2012) introduced Role eXtraction (RolX), an unsupervised method for automatically extracting structural roles from directed networks. It employs a mixed-membership strategy that distributes each node's role among the detected roles. Three steps—recursive feature extraction (ReFeX), feature grouping, and model selection—help RolX determine node roles. Kaslovsky developed the recursive feature extraction approach and a node role extractor (*GraphRole*) that we will employ in this research based on Henderson *et al.*'s papers (Kaslovsky, 2019). RolX uses a mixed-membership technique to mechanically extract roles from a graph. It also uses non-negative matrix factorization to approximate the node-feature matrix V:

$$V_{n\times f} \approx G_{n\times r} \times F_{r\times f} \tag{2.7}$$

where entries G_{ij} quantify the membership of node n_i in role r_j and entries F_{jk} specify how a membership in role r_j contributes to the value of feature F_{jk}. Given the number of roles, denoted by r, RolX applies. The rank r of this approximation is equal to the total number of roles. These two matrices efficiently compress V, supposing that node roles summarize node activity in the network.

2.4 Related Works

In this session we discuss related works that have being carried out in this domain. Our focus is on gene essentiality prediction in *Plasmodium falciparum* model organism, and the application of ML and network science techniques in essentiality studies.

2.4.1 GSMM Reconstruction: *Plasmodium falciparum* Model Organism

For the asexual blood stage of the malaria parasite *Plasmodium falciparum*, Carey *et al.* (2017) created a model known as iPfal17, which is a refined and expanded version of an earlier model known as iTH366 (Carey, Papin, and Guler, 2017). The updated model increased species- and stage-specificity and added more reactions, genes, and annotations. It included five compartments and had notes for blocked reactions as well as references for altered reactions. Their research emphasized the significance of precise metabolic network models for comprehending parasite activity and for medication discovery efforts. Overall, their research offered a fresh understanding of parasite biology and artemisinin resistance and suggested useful directions for choosing potential therapeutic targets.

In 2018, Abdel-Haleem *et al*. developed the iAM-Pf480 metabolic network, which is a carefully curated and quality-controlled GSMN model of *Plasmodium falciparum* (Abdel-Haleem *et al., 2*018). Using this model, they were able to explore the capabilities of malaria parasites to survive across the many phases of their complex life cycles. *P. falciparum* Malaria Parasite Metabolic Pathway (MPMP) Database (http://mpmp.huji.ac.il/), genome annotation from *Plasmodium* database (Plasmodb.org), and 332 main and review literature reference papers on biochemical and genetic characterization were used in the development of iAM-Pf480. The cytoplasm, endoplasmic reticulum, apicoplasts that resemble plastids, mitochondria, the Golgi apparatus, and lysosomes are all components of *PLASMODIUM falciparum*'s metabolic network. It includes 480 genes, 1083 activities, and 617 distinct metabolites. The gene-protein rule was found to be connected to 68% of enzymatic pathways and 480 genes. The essentiality predictions provided by iAM-Pf480 were compared to those made by the GSMN models iTH366 (Plata *et al., 2*010) and iPfal17 that were previously published. These comparisons were made using a wide range of experimentally verified targets (Carey *et al., 2*017). iAM-PF480 has a more advantageous genetic composition, a more comprehensive biochemical complement, and improved performance. The iAM-Pf480 was the GSMN model that was investigated for this investigation.

2.4.2 Machine Learning for Essentiality Prediction

Using local network topology, gene homologies, co-expression, and flux balance analysis, Plaimas *et al*. (2008) developed a machine learning technique to evaluate the essentiality of metabolic genes. This approach was used to assess essential enzymes in a metabolic network. The system was trained and validated with a dataset of knockout mutants of Escherichia coli and achieved high accuracy and precision of 0.93 and 0.9, respectively. The authors suggested that these features were sufficient for defining essentiality and could be applied to less specific media conditions. The approach was tested on predicted drug targets and yielded promising results, with several predictions supported by existing experimental evidence (Plaimas *et al., 2*008).

In research that was published in 2017, Nandi *et al*. created a unique support vector ML-based approach with the aim of finding significant metabolic genes in Escherichia coli K-12 MG1655. To create the most accurate ML model that is feasible, a strategy was created that made use of a well-balanced training dataset, an organism-specific genotype, phenotypic characteristics, and optimal parameter features. To enhance classification performance, flux-coupled metabolic features, which are subnetwork-based attributes, were used. The method that has been proposed is better than those that have been used in the past, and it demonstrates that significant genes have high codon use biases, remain unchanged across homologous bacterial species with high GC contents, and have high levels of gene expression that also serve as physiological flow modules in metabolism (Nandi *et al., 2*017).

They further improved their work in 2020 (Nandi *et al., 2*020) by suggesting a novel ML framework for predicting important genes in organisms with low experimental data. This pipeline is intended to be used for species such as yeast. The pipeline had three stages: dimensionality reduction, unsupervised selection of features, and a Laplacian SVM-based semi-supervised ML technique. The methodology was tested on both prokaryotes and eukaryotes, and it demonstrated a high level of accuracy despite containing just a minimum amount of labeled data (auROC > 0.85 with 1% labeled data). The proposed pipeline was also utilized for predicting crucial genes in organisms like Leishmania sp., for which there is presently inadequate data based on known experiments. The identified possible therapeutic targets for the development of medicines and vaccines against disease-causing parasites may be found by employing the predicted essential genes, which provide crucial hints for gene essentiality prediction.

In an attempt to leverage computational predictions to provide a prioritized shortlist for experimental validation, Azhagesan *et al.* (2018) undertook research to determine important genes across various species using network-based attributes. They found that prior methods for predicting essential genes, which combined sequence-based features with sequence, network, and biological information, did not adequately take into account network organization and the significance of network position in essentiality (Azhagesan *et al., 2*018). They suggested finding reliable network-based characteristics to enhance the prediction of important genes across species by capturing network structure. They examined the protein-protein interaction (PPI) networks of 27 different bacterial species and assessed how well network-based measures performed in comparison to sequence-based features and more traditional network metrics like degree centrality and clustering coefficient. According to their findings, 27 different species had an AUROC of 0.847 and a precision of 0.320. We enhanced the AUROC to 0.857 and the accuracy to 0.335 by adding sequence-derived features to the whole collection of network characteristics. This demonstrated that the suggested network properties outperformed previous approaches and were successful in determining important genes across a variety of species.

The performance was also enhanced by merging network-based features with sequence-based features. The authors used leave-one-species-out validation to confirm the efficacy of their characteristics. The study's overall goals were to understand the role of network structure in essentiality prediction and to create better techniques for categorizing essential genes using network-based criteria (M. Ashtiani *et al., 2*018).

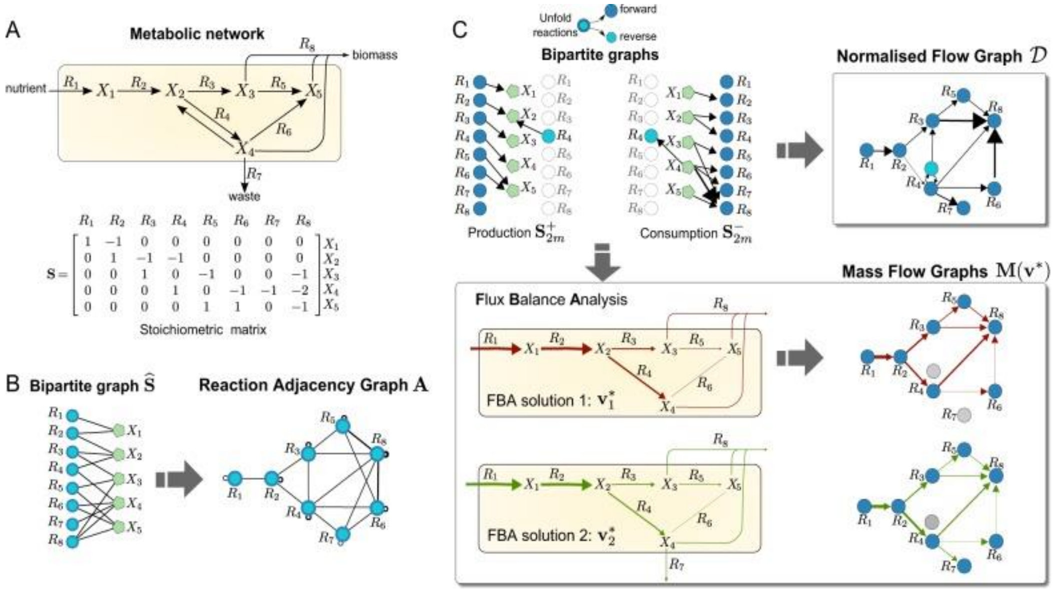
Aromolaran *et al*. (2021) suggested the use of ML methodologies as an adjunct to experimental methods for the purpose of locating important genes. The variables that affect the accurate computational prediction of relevant genes, as well as the benefits, drawbacks, and other contributing factors, are examined by Aromolaran *et al*. in their study from 2021. Due to the paucity of labeled data, the review emphasizes the limits of machine learning in predicting conditionally essential genes and concludes that careful feature selection and ML approaches are necessary for the accurate prediction of essential genes. The review also highlights the limitations of machine learning in predicting essential genes. They arrived at their conclusion by analyzing the performance of five distinct feature categories and discovered that topology features had the best discriminatory power, while gene ontology-based features were the most successful when it came to predicting essentiality. This led them to their conclusion.

2.4.3 Network-based ML for Gene Essentiality Prediction

In their review, Dusad *et al*. (2021) examined the link between FBA and graph-based investigations of metabolism as well as the complementary viewpoints they contribute to the comprehension of gene essentiality prediction in metabolic networks. In particular, the authors focused on how FBA and graph-based analyses of metabolism compared to other approaches. Although network science has the potential to offer insight on the emergent characteristics of global metabolic connections, FBA is a strong paradigm for forecasting metabolic fluxes based on network stoichiometry. They went further to establish the fact that there is a need to explore integrated approaches that combine flux optimisation with network science to better understand the complexity of metabolism as well as the potential for developing such hybrid pipelines (Dusad *et al., 2*021).

In an effort to understand global topological properties and the modularity structure of reaction graphs, Kim *et al.,* (2019) investigated the topological analysis of directed reaction centric metabolic networks of five different species (Kim *et al., 2*019). They examined directed reaction-centric graphs of metabolic networks in several microorganisms, ranging from prokaryotes to Saccharomyces Cerevisiae of the eukaryotes, using graph theory and centrality metrics. They examined several centrality measures, including those independent of the reaction nodes' degree, to better understand the topological roles played by certain reaction nodes. Their research sought to discover nodes with global topological significance and to comprehend local connectedness inside networks, which is crucial and indicates biological relevance. They also created a new statistic called the cascade number to evaluate the function of nodes in guiding mass flow. By estimating the proportions of crucial reactions predicted by FBA, they could connect the highly related reactions to their biological relevance. It needs to be seen if these graph topological properties may provide connection features that can be used to define relevant nodes, and if so, using ML models would make it simple for predicting essential nodes. This will make it simple to predict essential genes and find new medication targets, among other things which remain under research.

The terms "mass flow graphs" (MFG) and "normal flow graphs" (NFG) were proposed by Beguerisse-Diaz and colleagues in 2018 (Beguerisse-Díaz *et al., 2*018). The MFG represented a flux-dependent graph that may integrate environmental factors through reaction fluxes found using FBA, in contrast to the NFG, which represented an organism-wide metabolic organization. Their research addressed the drawbacks of prior graph designs by taking directionality into account, preventing over-representation of certain metabolites (known as pool metabolites that had great influence on the network topology thereby affecting biological inferencing), and taking environmental information into account. Using simulations of Escherichia coli and human hepatocyte metabolism, they showed the effectiveness of their method and how the graph topology changes in relation to the growth factors, offering fresh insights into the organization of the metabolism.Their work did not provide insight on how its properties may be used to help identify significant metabolic genes in a metabolic network, a topic that was later investigated by Freischem *et al.,* (2022). Figure 2.1 shows the flow diagram of the MFG which this study adopts.



A
Metabolic network
nutrient
biomass
waste
R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8
X_1 X_2 X_3 X_4 X_5
$\mathbf{S} = \begin{array}{c} \begin{array}{cccccccc} R_1 & R_2 & R_3 & R_4 & R_5 & R_6 & R_7 & R_8 \end{array} \\ \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & -1 & -1 & -2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 \end{bmatrix} \begin{array}{c} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{array} \end{array}$
Stoichiometric matrix
B
Bipartite graph $\widehat{\mathbf{S}}$
Reaction Adjacency Graph $\mathbf{A}$
C
Unfold reactions
forward
reverse
Bipartite graphs
Production $\mathbf{S}_{2m}^{+}$
Consumption $\mathbf{S}_{2m}^{-}$
Normalised Flow Graph $\mathcal{D}$
Mass Flow Graphs $\mathbf{M}(\mathbf{v}^*)$
Flux Balance Analysis
FBA solution 1: $\mathbf{v}_1^*$
FBA solution 2: $\mathbf{v}_2^*$

Figure 2.1 Flow diagram showing the NFG and MFG adopted from (Beguerisse-Diaz *et al.*, 2018)

Using graph-based methods and machine learning, Freischem *et al.*'s study from 2022 expanded on the work of Beguerisse-Daz *et al.* (2018) to improve essentiality prediction in metabolic genes. Their study centered on creating a machine learning pipeline that forecasts important genes using the mass flow graphs' connection properties (Freischem *et al., 2*022). In contrast to computationally expensive techniques like Flux Balance Analysis (FBA) and gene knockout simulations, they used the metabolic network's wild-type flux distribution to determine gene essentiality directly. They effectively predicted gene essentiality using data from Escherichia coli growth tests by building a mass flow graph to reflect flux distribution and training machine learning models based on graph node connections.

Their study is important because it helps to understand the minimum functional modules of an organism and has ramifications for biomedical and biotechnological applications. They don't require FBA solutions for deletion strains since their special machine learning pipeline, in combination with network science techniques, determines gene essentiality straight from the wild-type flux distribution. They used the Beguerisse-Daz *et al.* (2018) MFG-algorithm, projecting the flux distribution onto a mass flow graph, and using graph node connections to train machine learning models as a binary classification issue. They proved the precision and effectiveness of their method for identifying crucial genes through testing using data on Escherichia coli growth (Freischem *et al., 2*022).

When tested on the test set, the Random Forest model outperformed the other machine learning models in terms of overall accuracy, precision (82.5%), and recall (86.8%). This shows that there is enough information in the wild-type flux pattern to anticipate the need of metabolic genes. Overall, this work advances our knowledge of intricate biological systems and offers a potential framework for predicting gene essentiality.

2.4.4 Summary of Findings

Currently, to the best of our knowledge, there are no studies that explore and validate the application of FBA and prediction in network science coupled with ML techniques in gene essentiality prediction in eukaryotic *P. falciparum*, which our current research focus seeks to address.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Preamble

This chapter outlines the process used to collect the data, Flux Balance Analysis (FBA) and Mass Flow Graph Implementation (MFG), data processing and designing, and methodology for executing the experiment. We will consider the specifics of training and testing of the various machine learning techniques such as random forests, SVM, logistic regression, Naïve Bayes and Decision Tree determining their performance. The workflow the experiments is shown in Figure 3.1.

3.2 Dataset

In this section, we discuss the datasets that was used in this study, which is the genome-scale metabolic model of *Plasmodium falciparum* and its genomic content.

3.2.1 Genome-Scale Metabolic Model (iAM-Pf480)

As part of this work, the most recent GSMN model of *P. falciparum* (iAM_pf480) curated by Abdel-Haleem *et al.* (2018) that is publicly available on the BiGG, a knowledge base GSMN model (http://bigg.ucsd.edu/) was used. The *P. falciparum* GSMN model has 480 genes, 617 distinct metabolites, and 1083 reactions. Gene-protein-reaction (GPR) interactions involving 480 genes and 68% of all enzymatic processes was discovered in the model (Abdel-Haleem *et al., 2*018). The iAM_Pf480 is described in Table 3.1.

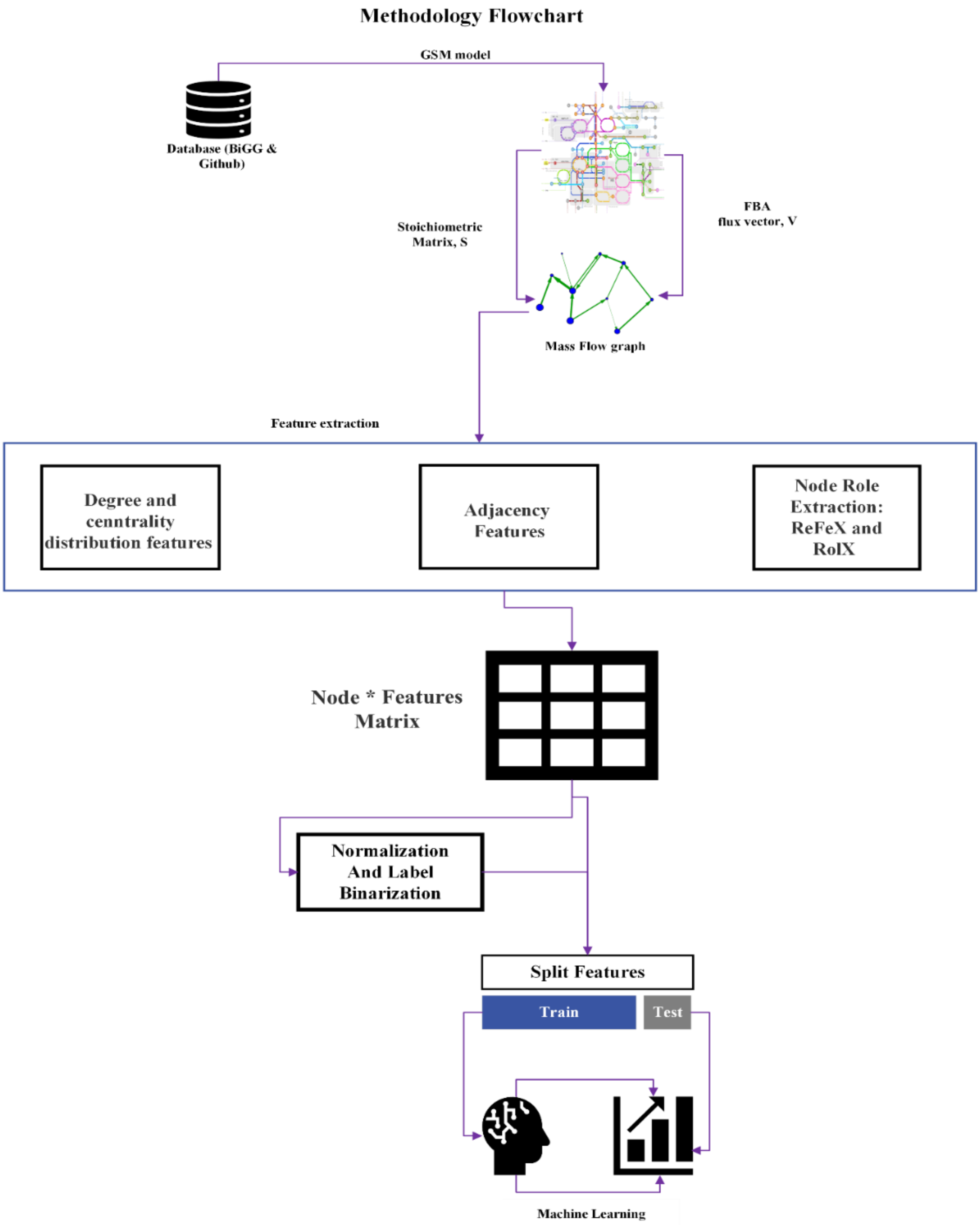


Figure 3.1: Workflow Diagram of Experiment

Table 3.1: Description of iAM_Pf480 Content

Metabolites		**905**
	Unique Metabolites	617
	Cytoplasm	531
	Apicoplast	109
	Golgi	45
	Mitochondria	82
	Endoplasmic Reticulum	26
	Lysosome	9
	Extracellular	107
Reactions		**1082**
	Gene-Associated Reactions (Metabolic & Transport)	738 (68%)
	Exchange Reactions	92 (9%)
	Non-Gene Associated React (Metabolic)	76 (7%)
	Non-Gene Associated React (Transport)	160 (15%)
	Demand and Sink Reaction	16 (1%)
Genes		**409**

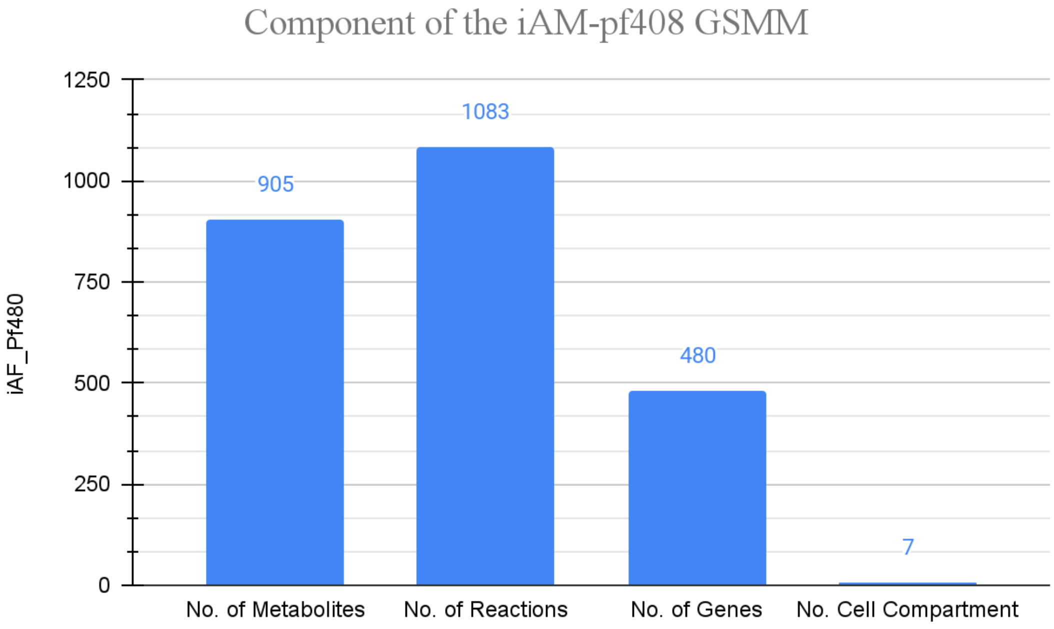


Figure 3.2: Reaction and gene counts in iAM-f480

In Figure 3.2, shows the total number of metabolites, the total number of reactions, the number of genes, and the total number of cellular compartments from which the reactions occur, which implies gene locations.

The cytosol, mitochondria, Golgi apparatus, endoplasmic reticulum, food vacuole, and apicoplast are the six different subcellular localizations that are represented in the iAM-Pf480 model, and the enzymes were fetched across all the stages of the development process of the model organism. The extracellular compartment is also included in this model. Previous existing models of the *Plasmodium falciparum* species iTH366 (Plata *et al., 2*010), iPfa (Chiappino-Pepe *et al., 2*021), and iPfal17 (Carey *et al., 2*017) were compared with iAM-PF480, but iAM-Pf480 was shown (Table 3.2) to span a greater variety of genetic content and have a bigger biochemical content. In addition to this, when compared to prior versions of the *P. falciparum* model that had been published, it demonstrated superior performance on golds standard dataset that was used in their research (Abdel-Haleem *et al., 2*018). Presently, iAM-Pf480 outperformed earlier *P. falciparum* GSMN models in terms of functionality and has a wider breadth of genomic material and a more extensive set of biochemical data, making it appropriate for the investigation.

Abdel-Haleem *et al*. (2018) evaluated the accuracy of the predictions and the capacity to correctly identify significant genes by comparing the iAM-Pf480 predictions to a list of empirically validated gene knockouts and drug-induced phenotypes in *P. falciparum*. The results showed that, in normal growth conditions, iAM-Pf480 had a 95% accuracy rate for single gene knockouts.

Table 3.2: GSMN models of *Plasmodium falciparum*

	iTH366	iPfa	iPf17	**iAM_pf480**
Reactions	1001	1066	1192	**1083**
Metabolites	915	1258	991	**617**
Genes	366	318	482	**480**
Biomass Component	51	73	82	**52**

3.3 Flux Balance Analysis (FBA)

FBA is one of the primary Linear Programming (LP) approaches utilized in the solution of GSMM/s Constraint-Based issues. It is a LP technique that computes the best steady-state flux distribution of a cell; the flux distribution specifies the cell phenotype (Freischem *et al., 2*022; Gatto *et al., 2*015; Sahu *et al., 2*021; Wu *et al., 2*016). A GSMN is used as input. The stoichiometric matrix and upper and lower constraints on reaction fluxes are used to generate a linear system of equations. In addition, the cell objective function Z is specified, which encodes the biological aim of the cell (Freischem *et al*., 2022). This is presumptively predicated on the idea that cell metabolism is optimized for maximum growth. FBA determines the solution vector to the following restricted optimization issue using linear programming:

$$Max\ Z = C^T V \tag{3.1}$$

$$subject\ to:$$

$$\frac{dy}{dt} = SV = 0 \tag{3.1a}$$

$$v_l \leq v \leq v_u \tag{3.1b}$$

where C encodes the cell objective function v_l and v_u are vectors containing the lower and upper limits on the fluxes of the reactions involved, respectively. Researchers are able to determine flux-flow of the cells under different environmental and genetic conditions by altering the reaction flux bounds (Martins Conde *et al., 2*016; Yasemi and Jolicoeur, 2021)

FBA has been applied primarily in the studies of gene essentiality prediction via performing single/double gene and/reaction essentiality in silico simulations.

3.3.1 FBA Implementation

Becker *et al.* (2007) created the COBRA toolbox, a popular software package that provides techniques for constraint-based development and analysis of GEMs in MATLAB, in response to the growth of GSMN modeling.

Ebrahim *et al.* (2013) introduced COBRA for Python (COBRApy) as part of the openCOBRA project, a community effort aimed at improving access to and promoting constraint-based research by making software openly accessible. COBRApy is a framework for object-oriented programming. It includes the capability for reading and writing COBRA models as SBML files, as well as executing FBA. COBRApy also offers an interface to linear optimization programs. This allows users to tackle FBA issues using current LP solvers such as GLPK or Gurobi (Gurobi, 2020). COBRApy's object-oriented architecture provides information about GEM reactions and metabolites readily accessible for various Flux base analysis. COBRApy (on 16gig RAM and a 64 bit window PC) was used in calculating the wild-type flux distribution vector using.

3.3.2 FBA Gene Essentiality Analysis

In order to compare our approach with outcomes of traditional FBA method, the study assessed the essentiality of reaction R_i by measuring its impact on the cell's growth rate when knocked out. To achieve this, COBRApy models were deployed to calculate and conduct single-reaction deletion simulations on the dataset.

The study assumes the objective of the biomass component as a maximization problem which is interpret as the cell’s aim of maximizing growth rate in the presence of glucose.

Assume the cell’s objective is given as

$$Z = F_{BIOMASS} \tag{3.2}$$

Define reaction’s impact on the objective (growth impact ratio) as.

$$\Delta Z_{R_i} = \frac{Z_{R_i}}{Z_{WT}} \tag{3.3}$$

Where Z_{R_i} is the optimal value of Z in the cell R_i is deleted/knocked out and Z_{WT} is the optimal value of Z in the wild type (Thus, without any knockout).

Using the growth impact ratio, ΔZ_{R_i}, define reaction essentiality as,

$$e_{R_i} = 1 - \Delta Z_{R_i} \tag{3.4}$$

Where;

$$e_{R_i} = \begin{cases} \geq 0.5 & Essential \\ < 0.5 & Nonessential \end{cases}$$

The study carried out individual reaction knockouts and ran the computationally demanding FBA for each knockout to ascertain the necessity of reactions in the genome-scale metabolic model (GEM). In order to compare the predictions with the outcomes of our machine learning (ML) approach, we noted the essentiality of each reaction that was represented in our dataset (equation 3.4). This gave us the opportunity to compare our method to the conventional FBA approach and evaluate the precision and efficacy of our method in predicting reaction essentiality over the full metabolic network.

3.4 Mathematical Theory of Mass Flow Graphs (MFG)

MFGs were first proposed by Beguerisse-Díaz *et al.*, (2018) as a way to map flux vectors onto a directed graph that can be analyzed with tools from network science. Each node in these graphs corresponds to a metabolic process, and two nodes are connected if they both utilize the same metabolite as either reactants or products. One of the best things about these reaction-centered graphs is that you don't have to get rid of pool metabolites, which are things like enzyme cofactors, ions, and other things that show up in many metabolic reactions. Pool metabolites are manually eliminated in other methods of graph construction to prevent the misleading connections brought about due to the high connectivity they form, which tend to dominate and affect the topological structure of the network.

In the MFG construction, these pool metabolites are not removed manually but they tend to map onto weak connections between graph nodes which makes their effectiveness on the overall connectivity much smaller. In order to build a MFG, the weight of the link between reactions R_i and R_j is defined as the total flow of metabolites generated by R_i and consumed by R_j. Mathematically, an MFG's adjacency matrix may be generated directly from the stoichiometric matrix S and a wildtype FBA solution vector. First, the flow vector is separated into 2 forward and reverse reactions (Freischem *et al.*, 2022):

$$V_{2m}^{*} = \begin{bmatrix} v^{*+} \\ v^{*-} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} abs(v) + v^{*} \\ abs(v) - v^{*} \end{bmatrix} \tag{3.5}$$

Then redefine the stoichiometric Matrix resulting from the GSMM construction as

$$S_{2m} = [S \ \ -S] \begin{bmatrix} I_m & 0 \\ 0 & diag(r) \end{bmatrix} \tag{3.6}$$

Where r, the reversibility vector with m dimensionality representing the reaction's reversibility as initialize during the GSMN construction, thus:

$$r_j = \begin{cases} 1 & if\ reaction\ j\ is\ reversible \\ 0 & otherswise \end{cases}$$

The resulting S_{2m} matrix is used to produce the production and consumption stoichiometric matrices as:

$$Consumption: \qquad S_{2m}^{+} = \frac{1}{2}(abs(S_{2m}) + S_{2m}) \qquad (3.6a)$$

$$Production: \qquad S_{2m}^{+} = \frac{1}{2}(abs(S_{2m}) + S_{2m}) \qquad (3.6b)$$

The flux vector of consumption and production is computed as:

$$j_i(v) = S_{2m}^{+} v_{2m}^{*} = S_{2m}^{-} v_{2m}^{*} \qquad (3.7)$$

So, considering a metabolite X_{ij} , $j_i(v)$ is the flux in which it is produced and consumed. In this case S_{2m}^{+} and S_{2m}^{-} are equal under steady conditions.

Hence the MFG adjacency matrix is computed as:

$$\boldsymbol{M(v^{*}) = (S_{2m}^{+} V^{*})^{T} J_{v}^{\tau} (S_{2m}^{-} V^{*})} \qquad \mathbf{(3.8)}$$

Where;

$V^{*} = diag(v_{2m}^{*})$

$J_v = diag\big(j(v^{*})\big)\ and\ \tau = the\ matrix\ pseudoinverse\ of\ J_v$

3.4.1 Binary Classification Problem on Mass Flow graphs

In supervised machine learning, the core objective of binary classification is to find patterns in observations of two classes and use those patterns to automatically classify unseen objects. To evaluate whether a gene is essential or not in the environment at hand, the study was focus developing automated classifiers that use growth metrics from knock-out tests and graph features of nodes in the Flux-weighted Reaction Centric Graph.

In this case, we have N pairs of data points, where each pair consists of a binary label y(i) indicating whether the gene is important or not and a p-dimensional feature vector $x(i)$ linked to the i^{th} gene/reaction. These feature vectors and labels are arranged into a class label vector (y) and a feature matrix (X).
Mathematically.

$$\big[\big(x^{(1)}, y^{(1)}\big), \qquad \big(x^{(2)}, y^{(2)}\big), \qquad \big(x^{(3)}, y^{(3)}\big), \dots, \big(x^{(N)}, y^{(N)}\big)\,\big] \qquad (3.9)$$

where $x^{(i)} \in R^{p}$ is a $p - dimensional$ vector features associated with the reaction/genes and $y^{(i)} \in \{0,1\}$ is the class label that is associated with the class label of the reaction or gene. We regard non-essential genes as the negative class (0) and essential genes as the positive class (1) without loss of generality.

All of the feature vectors are included in the feature matrix, X, and the matching class labels are represented by the label vector y. The feature vectors and labels are assembled into a feature matrix X and a vector of class labels y:

$$\begin{aligned} X &= \left[x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(j)}\right] \\ y &= \left[y^{(1)}, y^{(2)}, y^{(3)}, \dots, y^{(j)}\right] \end{aligned} \tag{3.10}$$

Using this data, a classification algorithm is trained to discover patterns in the input data and predict labels in the output data. Once trained, the system may use the feature vectors of unseen samples to automatically identify their labels.

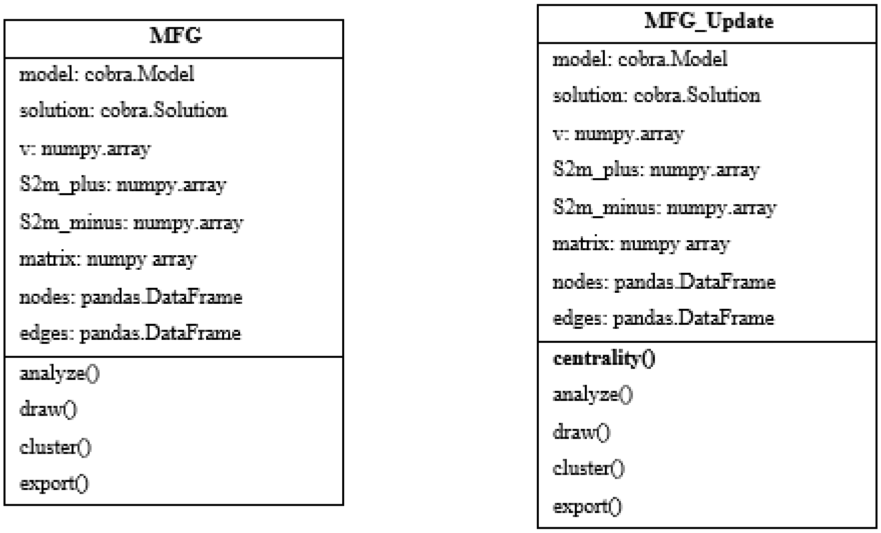
There are several categorization model types that may be used, and each one makes a different assumption about how the feature space is shaped. Support vector machines, decision trees, neural networks, and logistic regression are examples of common models that will be used. The job and dataset at hand determine which model should be used. In a typical machine learning pipeline, cross-validation studies are performed alongside model training to minimize overfitting and to conduct model selection, making sure the selected model works well on untrained datasets.

3.5 Graph Feature Extraction

In research conducted by Freischem *et al*., (2022), they implemented the mass flow graph theoretical framework that was proposed by Beguerisse-Díaz *et al*., (2018). They developed MFGpy, a python package for the automatic generation, analysis, and visualization of MFGs from a COBRA model. The GSMN model (iAM_Pf430) that was downloaded from the BiGG database was utilized to construct the graph network using this software.

The study adopted MFGpy, a Python package created by Freischem *et al.* in 2022. According to their study (Freischem, Barahona, and Oyarzn, 2022) the package merges COBRApy with Mass Flow Graphs (MFGs). For automating MFG generation from GSMMs (Genome-Scale Metabolic Models), this methodology provides a number of functionalities. Additionally, it makes it easier to analyse MFGs and export them as numpy and CSV files for future study.

Table 3.3: UML of MFG python class



MFG
model: cobra.Model solution: cobra.Solution v: numpy.array S2m_plus: numpy.array S2m_minus: numpy.array matrix: numpy array nodes: pandas.DataFrame edges: pandas.DataFrame

analyze()
draw()
cluster()
export()

MFG_Update
model: cobra.Model solution: cobra.Solution v: numpy.array S2m_plus: numpy.array S2m_minus: numpy.array matrix: numpy array nodes: pandas.DataFrame edges: pandas.DataFrame

centrality()
analyze()
draw()
cluster()
export()

The study updated the MFGpy class model to include an automatic centrality feature extraction from the resulting MFG to aid in further graph analysis. Hence, in the machine learning stage, the extracted centrality features—clustering coefficient, proximity centrality, betweenness centrality, and degree centrality—can be exported as a node feature matrix and stored as a CSV file.

The COBRA model that corresponds to the MFGpy is stored in the *model*. This allows running the GSMM through a Flux Balance Analysis (FBA). The production and consumption stoichiometric matrices are kept in *'S2m_plus'* and *'S2m_minus'*, respectively, while the FBA analysis's output flux vector is stored in *'v'*. *'v'*, *'S2m_plus'*, and *'S2m_minus*' are used to create the adjacency matrix as indicated in equation (3.6).

MFGpy also enables network visualisation and clustering analysis using the NetworkX Python library (Hagberg *et al.*, 2008). MFG then include *export()* which allows the resulting graph to be exported as a numpy file for additional analysis as well as a CSV file containing nodes and edge weights Table. Graph visualisation tools like Gephi 0.10 and Cytoscape (Paul Shannon *et al.*, 1971) can be used to visualise the CSV file.

The study performed the FBA of iAM-Pf480 to calculate the wild-type flux vector using COBRApy considering glucose as energy source; then implemented the MFG algorithm with a written code in python 3.9.7 language and NetworkX library on a Linux operating system, AMD core, 750 HD and 16 Gigabyte RAM. The flux-weighted graph created using MFG contains 505 reactions with 6217 edges and is viewed using Gephi 0.10 software as shown in Figure 3.2.

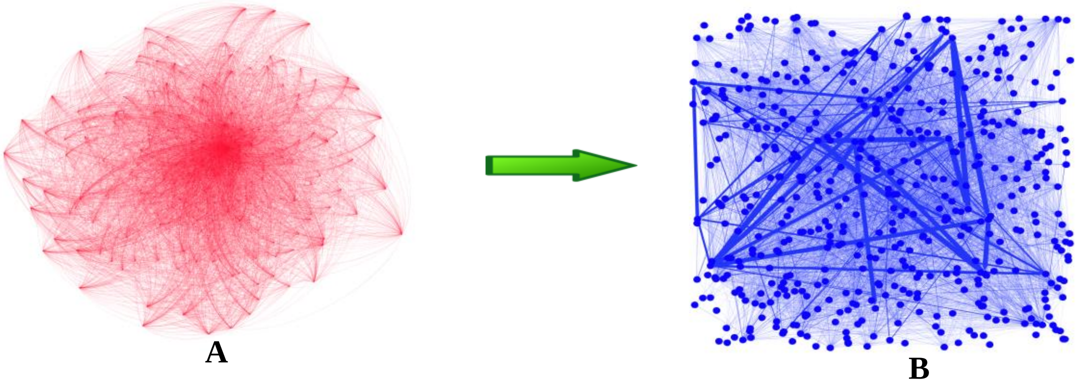


Figure 3.2: Mass Flow graph of iAM-Pf480 view in Gephi 0.10

(A. Shows flux-weighted graph of iAM_Pf480 GSMM (when the graph is unweighted) and B. shows the graph with it edge weight.)

3.5.1 Centrality-based Features

From the flux-weighted network of iAM_Pf480, the researcher extracted six topology-based metrics, including PageRank, PageRank Percentage, Betweenness centrality, Closeness centrality, Clustering Coefficient, and degree using *MFG_updatepy* written in python which were exported in the form of a CSV file. The feature matrix has 330 rows and 6 columns, each row representing one of the 330 reactions and each column indicating one of the feature values associated with that reaction. After that, we used this feature matrix as an input for the machine learning model to train and make prediction on gene essentiality.

3.5.2 Node Role Analysis

In this section, we discuss features that are extracted based on the role of nodes in the network graphs.

A. Recursive Feature Extraction Algorithm (ReFeX) and Role eXtraction (RolX)

Regional features of nodes are computed recursively by the Recursive Feature eXtraction algorithm (ReFeX) as discussed in chapter 2. Neighbourhood features, which include local and egonet properties, are computed to initialize the algorithm. The egonet of a node consists of the node itself, all its neighbours, and all edges within this group of nodes; ReFeX also considers the egonet's incoming and outgoing edges. Node degree is measured by local features. They include weighted in-, out-, and total degrees for a weighted digraph. Egonet features are the weighted, directed number of edges that are present inside, entering, and exiting the egonet. The advantage of ReFeX features is its scalability and effectiveness in discovering regional information in large network graphs.

B. Implementation of ReFeX and RolX

The ReFeX and RolX algorithms both was implemented as *GraphRole* module that is available via the PyPi project (Kaslovsky, 2019) which was adopted for research. The development of these algorithms, which is covered in more depth in Chapter 2 of the dissertation proposed by Henderson and colleagues (Henderson *et al.*, 2011, 2012). *GraphRole* includes methods that implements **Re**cursive **Fe**ature **eX**tractor for the ReFeX algorithm and **Rol**e **E**xtractor for the RolX algorithm when called on a graph.

The ReFeX and RolX methodologies depend on graph definitions used in NetworkX, which is a Python programme developed for the generation, modification, and investigation of complex networks. The multi graph definitions generated by NetworkX are used as input by the GraphRole module throughout the process of implementing the ReFeX and RolX algorithms (Kaslovsky, 2019). The output that is generated by these algorithms are stored as a data frame file which is exported as a ReFeX and RolX feature matrix in csv. The experiment resulted in extracting a ReFeX feature matrix of 330 rows reactions and 31 feature columns whereas for RolX, the study extracted a feature matrix of 330 rows and 5 columns (From the Flux-weighted reaction-centric graph resulting from Equation 3.8).

3.5.3 Adjacency Features

As a result of the large number of reactions in FBA solution vectors that do not transport flux, these reactions are mapped onto disjointed nodes in the Mass Flow Graphs, which means that they are non-essential genes. The adjacency matrix resulting from the MFG can be used as a node feature matrix and used to train the machine learning models testing their predictive power in essentiality prediction (Freischem *et al.*, 2022).

Mathematically:

Given the adjacency matrix M_m, of $(m \times m)$ size of metabolites m, let the nonzero nodes be k, all zero nodes are removed from M_m to form $(m - k)$ a reduced M_k. Hence the feature matrix X in formed by the form.

$$X_m = [M_k \; M_k^T] \tag{3.11}$$

In all, applying *GraphRole* on the graph, it used a mixed-membership assignment strategy to group the nodes into five separate roles $(r1, r2, r3, \ldots, r5)$. These roles are denoted by the letters r1, r2, r3, …, and r5. A percentage number, representing the node's contribution to each role, was ascribed to each individual node. In addition, topological parameters including PageRank, PageRank Percentile, Degree, Clustering Coefficient (CoC), Betweenness Centrality (BC), and Closeness Centrality (CC) were used in the investigation. In addition, the adjacency features (F1, ..., F1010) were recovered from the adjacency matrix that was produced by the graph that was created.

3.6 Essentiality Labelling

The research study made use of the OGEE database to extract *Plasmodium falciparum* essential genes saved as a csv file. The **O**nline **GE**ne **E**ssentiality (OGEE) database is a comprehensive source that specializes in essentiality data. It acts as a database of essential and non-essential genes that have undergone experimental verification. These genes have been collected from numerous sources and acquired via a variety of experimental techniques (Gurumayum *et al.*, 2021).

The OGEE database compiles information from several research papers that cover a variety of species and experimental techniques. It comprises data on the relevance of certain genes for the survival and functioning of the organism gleaned via genetic knockout tests, RNA interference (RNAi) investigations, transposon mutagenesis, and other approaches(Hagberg *et al.*, 2008).

A collection of experimentally confirmed essential genes unique to *Plasmodium falciparum* from the OGEE database was downloaded and used in this research study. This data provides a critical starting point for additional research and analysis aimed at elucidating the fundamental biological mechanisms and possible therapeutic targets present in the *Plasmodium falciparum* genome. This essentiality information was used as essentiality labels for the reaction nodes following Gene-Protein Reaction (GPR) rules that are contained in the genome scale metabolic model (iAM_Pf480).

3.7 Data Preprocessing

In MFG algorithm, poorly linked reactions are automatically removed using the flux vector from the flux balance analysis (FBA), thus, showing their non-essential status (Beguerisse-Díaz *et al.*, 2018). As a result, the iAM_Pf480 Genome-Scale Metabolic Model (GSMM)'s initial 1082 reactions were reduced in our experiment to 505 reactions. It produced a graph with 6217 edges. We also removed reactions from the graph which are not linked to any genes (non-gene associated genes). Since there are no gene connections for these reactions, their essentiality or non-essentiality cannot be identified. Consequently, we were able to identify 330 reactions, of which 258 were essential and 72 were not (Figure 3.3).

Hence, dataset is made up of 78.2% essential reactions (175 essential reactions) and 21.8% essential reactions (155 non-essential reactions).

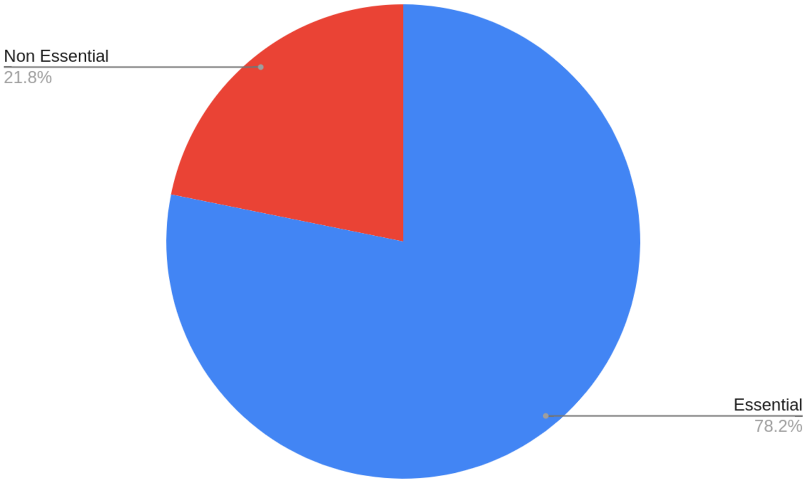


Figure 3.3: Pie chart of Essential and Nonessential reactions

3.8 Data Normalization

In machine learning, normalization is a crucial preprocessing step, especially when working with classification algorithms that depend on calculating the distances between feature vectors. The size of the features may have an impact on distances like Euclidean or cosine distances, which can cause bias or portray the real connections between samples incorrectly.

Before training the classification model, we conducted feature matrix X normalization to solve this problem and avoid issues brought on by feature scaling. By bringing all features to a comparable scale, the normalization procedure seeks to prevent any dominance or distortion brought on by different feature value ranges.

In this scenario, a particular normalization technique that protects the feature matrix's sparse nature was used. We concentrated just on scaling the features to have unit variance rather than removing the feature mean and scaling to unit variance (mean normalization). This method achieves the necessary scaling effect while preserving the original distribution and connections between the features. Each element in the feature matrix X is divided by the standard deviation of the accompanying feature j to normalize the matrix shown in equation (3.12). A normalized entry x_{ij} that represents the feature's scaled value is created because of this operation. Based on the training data, the standard deviations σ_j are determined, and the normalization factors are saved for use in reliably converting additional input data in the future.

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sigma_j} \tag{3.12}$$

We reduce any bias or distortion that can be brought about by different feature scales by using this normalization procedure. As a result, the classification algorithm is no longer swayed by the scales of the features individually but instead concentrates on their relative relevance and linkages. The resilience and efficacy of the classification model in correctly identifying the patterns and producing accurate predictions are enhanced by this normalization phase.

3.9 Machine Learning Models

The study developed a machine learning pipeline as shown in **Figure 3.4** in order to train binary classifiers that can predict the essentiality labels based on characteristics collected from the mass flow graphs, using python programming language and Scikit-learn library. Various ML algorithms such Support Vector Machine (SVM), Logistic Regression (LG), Random Forest (RF), Decision Tree (DT), k-Nearest Neighbourhood (kNN) and Naive Bayes (NB) were deployed and their performance on the various datasets were evaluated. The study looked at the best performing ML algorithm and the features sets that gave the highest accuracy of prediction.

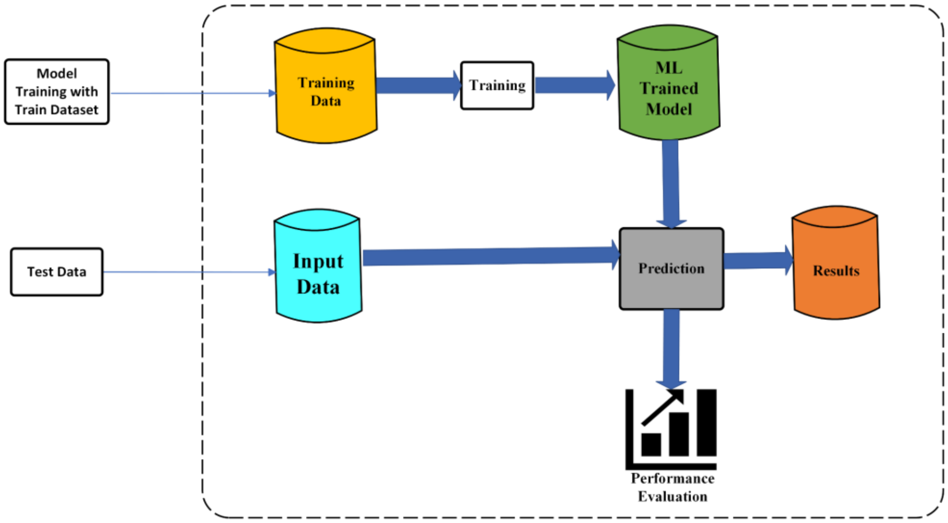


Figure 3.4: Machine Learning Pipeline

i. Support Vector Machine (SVM) Binary Classifier

A Support Vector Machine (SVM) finds a hyperplane which separates the two classes in the feature space. It aims to maximize the margin that separates the classes which reduces the risk of false classification. SVMs are memory efficient and versatile classifiers which can be used for binary as well as multi-class classification (Shmilovici, 2005).

ii. Logistic Regression Binary Classifier

The objective of the classification technique known as logistic regression is to establish a connection between feature frequencies and the likelihood of a specific outcome. In contrast to linear regression, which produces a straight line to represent the class probability, logistic regression produces a sigmoidal curve. The sigmoid function, which generates an S-shaped curve with a range of 0 to 1, transforms discrete or continuous numerical data (x) into a single numerical value (y), determines this curve. The main benefit of this strategy is that probabilities are restricted to the range between 0 and 1 (they cannot be less than 0 or larger than 1). It can either be binomial, with just two potential results, or multinomial, with three or more $y = \frac{1}{1-e^{x}}$ outcomes conceivable (Tiwari, 2020). Logistic Regression uses a logistic sigmoid function as transformation to linear output labels to model a binary output variable. The resulting outputs are between 0 and 1 and are interpreted as the probability of a sample to be in class 1. Maximum likelihood estimation is used to find weights that maximize the probability of the data.

iii. Random Forest Binary Classifier

The Random Forest method is a widely used machine learning algorithm for classification and regression tasks (Schonlau and Zou, 2020). It creates a "forest" of trees by combining different decision trees, resulting in a random subset of characteristics. The trees train using various subsets of training data, and the final prediction is made by combining the data from each tree's unique forecasts. The most frequent prediction is selected for classification tasks, while the average or weighted average is used for regression tasks. Random Forest can handle high-dimensional datasets with many attributes, recording intricate interactions and non-linearities between characteristics and target variables (Donges, 2021; Schonlau and Zou, 2020).

It also manages missing values without imputation and offers built-in feature significance measurements for feature selection and understanding of underlying connections. However, it can be computationally costly to train many trees, difficult to understand due to its intricacy and ensemble nature and may be biased towards the dominant class in unbalanced class distributions. Despite these drawbacks, Random Forest remains a useful tool in many machine learning applications due to its ability to manage complicated data and provide reliable predictions (Aleryani, Wang, and de la Iglesia, 2020).

iv. k-Nearest Neighbourhood (kNN) Binary Classifier

Originally created by Evelyn Fix and Joseph Hodges in 1951 and then enhanced by Thomas Cover, the k-nearest neighbours' algorithm (k-NN) is a non-parametric supervised learning technique (Cover and Hart, 1967). For both classification and regression problems, it is a straightforward and understandable machine learning technique. k closest neighbours in the training data are used by the kNN algorithm to identify the class or value of a new data point. How many neighbours are considered depends on the value of k. The new data point is classified by being given the class that has most of the k closest neighbours. As the projected value in regression, the k closest neighbours' values are averaged or weighted to provide an average. Assumptions concerning the distribution of the underlying data are not made by kNN since it is non-parametric. But since it involves figuring out how far every training sample is from every new data point; it may be computationally costly for huge datasets.

v. Decision Tree Binary Classifier

A Decision Tree is a well-liked machine learning model that is utilized for both classification and regression applications. It provides a structure that resembles a flowchart, where each leaf node is a class label or a projected value, and each inside node is a decision based on a particular characteristic. Decision trees divide the data depending on the attributes that divide the dataset most effectively, trying to minimize impurity or maximize information gain. The split criteria, like Gini impurity or entropy, rely on the algorithm being utilized. Decision trees have the benefit of being comprehensible and capable of capturing non-linear connections. However, they are prone to overfitting and might be sensitive to little variations in the training data (Tiwari, 2020).

vi. Naive Bayes Binary Classifier

Naive Bayes is a probabilistic machine learning method Based on Bayes' theorem. It is often used for classification jobs, notably spam filtering and text categorization. Naive Bayes makes the strong but more straightforward assumption that the characteristics are conditionally independent given the class label. Given the feature values, it calculates the likelihood of each class and chooses the class with the greatest probability as the predicted class. Even with a lot of features, Naive Bayes is computationally efficient. However, it may struggle with unusual occurrences or classes with unbalanced data, and its performance may suffer when the independence assumption is broken (Camacho *et al., 2*018).

3.10 Evaluation Metrics

We measured the performance of ML binary classifiers using the evaluation metrics discussed below:

i. Accuracy

Accuracy is one of the performance matrices calculated using the confusion matrix. It accounts for the percentage of outcomes that have been predicted correctly.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.13}$$

where TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative.

Accuracy is the most suiTable for cases of perfectly balanced data that must prove misleading in situations where our data is imbalanced.

ii. Precision

Precision is a measure of how many of the predicted positive outcomes are actually positive. It shows how many correct positive predictions there are compared to the total number of positive predictions.

$$Precision = \frac{TP}{TP + FP} \tag{3.14}$$

iii. Recall/Specificity

Recall shows how many of all the actual positive values are also predicted to be positive. It shows how many correct positive predictions there are compared to how many positive cases there are in the entire dataset.

$$Recall = \frac{TP}{TP + FN} \tag{3.15}$$

iv. F1 Score

It is a balance between recall and accuracy. Its interval is [0,1]. This statistic often informs us of the classifier's precision (number of cases properly classified) and robustness (absence of significant number of missed instances).

$$F1\ Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \tag{3.15}$$

3.11 Summary of Experimentation

A weighted reaction centric graph with 505 reactions as nodes and 6217 edges was constructed from the Genome-scale metabolic model. Amidst this reaction may include reactions that are not associated with any gene and will not have any essentiality label.

Hence, we dropped all these reactions after extracting the graph features resulting is 330 reactions remaining.

ReFeX features (ReFeX), RolX features (RolX), Adjacency matrix features, and Topological/Centrality features are four separate sets of response features that were extracted from the graph. These feature sets were pooled, and many machine learning algorithms were tested and trained using them. The objective was to evaluate how well these algorithms performed in foretelling the essentiality of nodes which have been labelled. Based on the matching forecast, every node was classified as either vital or non-essential. Table 3.4 reports details on the various data sets that were extracted from the graph.

Table 3.3: Various feature sets used in Experiment.

Dataset	No of Features
RolX	5
ReFeX	31
Topological Features	6
Adjacency Features	1010
ReFeX&RolX	36
Topology&ReFeX	37
Topology&RolX	11
Topology&ReFeX&RolX	42

3.11.1 Computational Power

All of the experiments in this project were carried out in Python 3 and made use of a variety of libraries, including scikit-learn (Pedregosa *et al.*, 2011), networkX (Hagberg *et al.*, 2008), MFG_updatepy (a modified version of MFGpy (Freischem *et al.*, 2022) for automated generation of MFG graphs and their centrality featutures), GraphRole (used for automated Recurrent Feature Extraction and node role analysis) (Kaslovsky, 2019), and the COBRApy (Ebrahim *et al.*, 2013) libraries. The algorithms and scripts were run on a personal computer with an AMD Core CPU running at 2.70GHz and 16 GB of RAM. The script form MFG implementation used for this project is included in the appendix B, and we intend to make it publicly available on GitHub at https://github.com/stephen-bin.

CHAPER FOUR

RESULTS AND DISCUSSION

4.1 Preamble

In this chapter, we discuss results that we obtained from our experiments. We show the resulting flux-weighted reaction centric graph and evaluate the performance of ML algorithms across the graph-based features that we extracted in predicting gene essentiality.

The iAM-Pf480 GSMM used in the flux-weighted reaction-centric (FWRC) graph was adopted from BiGG database as a systems biology mark-up language (SBML) file and their essentiality labels were acquired from Ogee databased as a csv file and used train a supervised machine learning model to predict reaction essentiality. We utilized the iAM-Pf408 GSM Model to create the dataset. The resulting FWRC graph is shown in Figure 4.1.

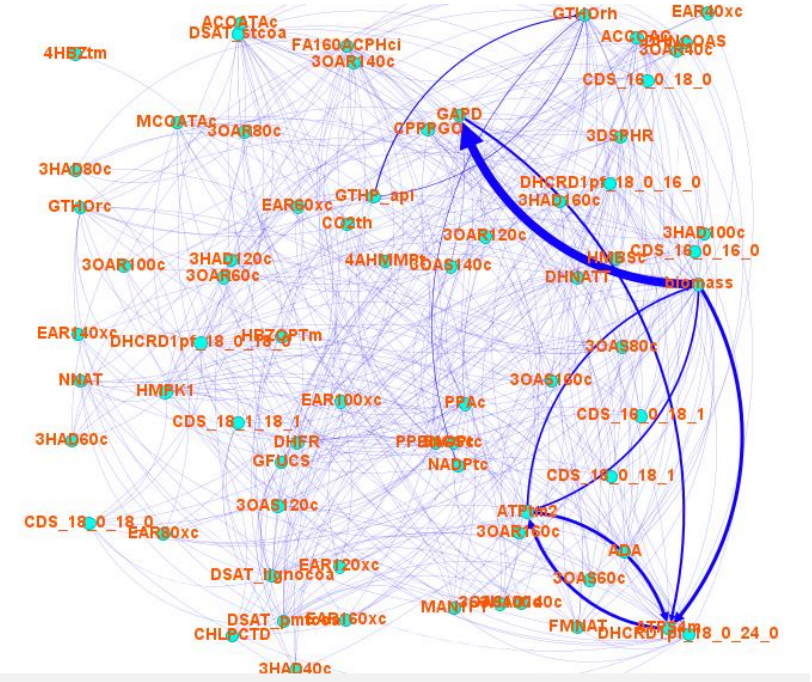


Figure 4.1: FWRC graph of iAM-Pf480 model viewed with Gephi 0.10

4.2 Results

To begin, we trained six distinct models for binary classification. To ensure optimal performance, we utilized 5-fold cross-validation to fine-tune the hyperparameters of each model. Cross-validation involves dividing the data into five subsets, or folds, and training the model on four folds while evaluating its performance on the remaining fold. This process is repeated five times, each time using a different fold as the evaluation set. By doing so, we obtain a more reliable estimate of the model's performance. We evaluated the performance of the models using five commonly used evaluation metrics: Area under the Receiver Operating Characteristic curve (AuROC), Accuracy, Precision, Recall, and F1-score. These metrics provide valuable insights into the model's ability to correctly classify the data. Considering the imbalanced nature of the datasets, where one class has significantly more instances than the other, we employed a weighted average record for values of precision, recall, and the F1-score. This weighting accounts for the uneven distribution of classes, giving equal importance to both classes and ensuring a balanced evaluation of the model's performance. The results of the model evaluations, including the optimized hyperparameters, are summarized in Table 4.1. These results reflect the performance of the models trained on 80% of the available reactions. The remaining 20% of reactions were reserved as a held-out/test set, allowing for an unbiased assessment of the models' generalization ability on unseen data.

Table 4.1: Performance evaluation of ML methods on various feature sets

A. Adjacency Features

ML Model	AuROC	Accuracy	Precision	Recall	F1 - Score	Accuracy 5-Cross Val (5)
LogReg	0.41	0.64	0.57	0.64	0.6	[0.79, 0.79, 0.79, 0.77, 0.77]
KNN	0.48	0.74	0.59	0.74	0.66	[0.68, 0.64, 0.77, 0.77, 0.76]
Random Forest	0.52	0.73	0.67	0.73	0.69	[0.77, 0.74, 0.74, 0.76, 0.76]
SVM	0.5	0.77	0.6	0.77	0.67	[0.79, 0.79, 0.79, 0.77, 0.77]
Decision Tree	0.51	0.68	0.66	0.68	0.67	[0.80, 0.77, 0.62, 0.71, 0.74]
Naive Bayes	**0.56**	**0.79**	**0.76**	**0.79**	**0.73**	**[0.77, 0.62, 0.79, 0.77, 0.77]**

B. Topological Features

ML Model	AuROC	Accuracy	Precision	Recall	F1 - Score	Accuracy 5-Cross Val (5)
LogReg	0.46	0.71	0.59	0.71	0.64	[0.79, 0.79, 0.79, 0.77, 0.77]
KNN	0.58	0.74	0.71	0.74	0.72	[0.82, 0.74, 0.73, 0.79, 0.71]
Random Forest	**0.69**	**0.82**	**0.8**	**0.82**	**0.81**	**[0.85, 0.76, 0.77, 0.74, 0.77]**
SVM	0.5	0.77	0.6	0.77	0.67	[0.79, 0.79, 0.79, 0.77, 0.77]
Decision Tree	0.59	0.77	0.74	0.77	0.75	[0.71, 0.74, 0.73, 0.77, 0.74]
Naive Bayes	0.47	0.68	0.61	0.68	0.64	[0.73, 0.80, 0.79, 0.77, 0.77]

C. ReFeX

ML Model	AuROC	Accuracy	Precision	Recall	F1 - Score	Accuracy 5-Cross Val (5)
LogReg	0.62	0.82	0.82	0.82	0.78	[0.79, 0.67, 0.76, 0.77, 0.77]
KNN	0.57	0.77	0.73	0.77	0.73	[0.77, 0.74, 0.68, 0.77, 0.77]
Random Forest	**0.69**	**0.85**	**0.85**	**0.85**	**0.83**	**[0.85, 0.76, 0.74, 0.74, 0.77]**
SVM	0.53	0.79	0.83	0.79	0.71	[0.79, 0.79, 0.79, 0.77, 0.77]
Decision Tree	0.6	0.82	0.85	0.82	0.77	[0.67, 0.76, 0.76, 0.76, 0.79]
Naive Bayes	0.63	0.58	0.74	0.58	0.61	[0.44, 0.58, 0.39, 0.5, 0.56]

D. RolX

ML Model	AuROC	Accuracy	Precision	Recall	F1 - Score	Accuracy 5-Cross Val (5)
LogReg	0.5	0.77	0.6	0.77	0.67	[0.79, 0.79, 0.79, 0.77, 0.77]
KNN	0.49	0.65	0.64	0.65	0.65	[0.74, 0.82, 0.76, 0.68, 0.59]
Random Forest	0.55	0.77	0.72	0.77	0.72	[0.79, 0.76, 0.79, 0.77, 0.74]
SVM*	**0.56**	**0.79**	**0.76**	**0.79**	**73**	**[0.77, 0.79, 0.79, 0.77, 0.77]**
Decision Tree	0.49	0.76	0.56	0.76	0.67	[0.74, 0.76, 0.80, 0.77, 0.77]
Naive Bayes	0.49	0.76	0.59	0.76	0.67	[0.79, 0.71, 0.80, 0.62, 0.77]

E. ReFeX and RolX

ML Model	AuROC	Accuracy	Precision	Recall	F1 - Score	Accuracy 5-Cross Val (5)
LogReg	0.66	0.83	0.83	0.83	0.8	[0.79, 0.67, 0.76, 0.77, 0.77]
KNN	0.59	0.8	0.79	0.8	0.76	[0.77, 0.74, 0.68, 0.77, 0.77]
Random Forest	**0.68**	**0.83**	**0.82**	**0.83**	**0.81**	**[0.83, 0.76, 0.74, 0.74, 0.74]**
SVM	0.5	0.77	0.6	0.77	0.67	[0.79, 0.79, 0.79, 0.77, 0.77]
Decision Tree	0.62	0.82	0.82	0.82	0.78	[0.74, 0.77, 0.74, 0.77, 0.76]
Naive Bayes	0.65	0.61	0.75	0.61	0.64	[0.44, 0.56, 0.39, 0.5, 0.56]

F. TopologyReFeX

ML Model	AuROC	Accuracy	Precision	Recall	F1 - Score	Accuracy 5-Cross Val (5)
LogReg	0.62	0.77	0.75	0.77	0.76	[0.89, 0.74, 0.76, 0.79, 0.76]
KNN	0.62	0.82	0.82	0.82	0.78	[0.79, 0.74, 0.68, 0.77, 0.77]
Random Forest	**0.69**	**0.85**	**0.85**	**0.85**	**0.83**	**[0.82, 0.77, 0.74, 0.76, 0.76]**
SVM	0.53	0.79	0.83	0.79	0.71	[0.79, 0.79, 0.79, 0.77, 0.77]
Decision Tree	0.63	0.79	0.76	0.79	0.77	[0.74, 0.74, 0.73, 0.76, 0.77]
Naive Bayes	0.63	0.58	0.74	0.58	0.61	[0.44, 0.62, 0.39, 0.52, 0.56]

G. TopologyRolX

ML Model	AuROC	Accuracy	Precision	Recall	F1 - Score	Accuracy 5-Cross Val (5)
LogReg	0.46	0.71	0.59	0.71	0.64	[0.79, 0.79, 0.79, 0.77, 0.77]
KNN	0.66	0.8	0.78	0.8	0.79	[0.82, 0.76, 0.73, 0.79, 0.71]
Random Forest	**0.67**	**0.82**	**0.8**	**0.82**	**0.8**	**[0.82, 0.71, 0.74, 0.73, 0.77]**
SVM	0.52	0.77	0.72	0.77	0.7	[0.79, 0.79, 0.79, 0.77, 0.77]
Decision Tree	0.61	0.8	0.78	0.8	0.77	[0.76, 0.77, 0.76, 0.77, 0.77]
Naive Bayes	0.48	0.67	0.63	0.67	0.65	[0.73, 0.79, 0.73, 0.65, 0.70]

H. TopologyReFeXRolX

ML Model	AuROC	Accuracy	Precision	Recall	F1 - Score	Accuracy 5-Cross Val (5)
LogReg	0.61	0.76	0.74	0.76	0.74	[0.89, 0.74, 0.76, 0.79, 0.76]
KNN	0.59	0.8	0.79	0.8	0.76	[0.79, 0.74, 0.68, 0.77, 0.77]
Random Forest	**0.66**	**0.83**	**0.83**	**0.83**	**0.8**	**[0.83, 0.72, 0.74, 0.74, 0.77]**
SVM	0.53	0.79	0.83	0.79	0.71	[0.79, 0.79, 0.79, 0.77, 0.77]
Decision Tree	0.63	0.79	0.76	0.79	0.77	[0.74, 0.77, 0.73, 0.77, 0.77]
Naive Bayes	0.65	0.61	0.75	0.61	0.64	[0.42, 0.56, 0.39, 0.5, 0.56]

The study aimed to find the best model for binary classification tasks and the best dataset on which this model performs after model training, hyperparameter optimisation, and prediction. The Table above shows the experimental results of the performance of each ML model on the various Feature sets.

4.3 Discussion

In this section, we discuss the performance of the various ML models across all the datasets used in the experiment.

A. Adjacency Features (Table 5A)

We extracted the adjacency matrix from the mass flow graph and used it as 330 nodes by 1010 features matrix traced to the consumption and production stoichiometric matrix (Chapter 3 Equation 3.11). We observed that the ML models had poor predictive power on this dataset. Naïve Bayes achieved the highest performance with auROC of 0.56, accuracy and f1-score, 0.79 and 0.73 respectively, and this is probably due to its ability to work best on high dimensional dataset. Random Forest performed reasonably well with an auROC and accuracy of 0.52 and 0.73 respectively. The accuracy in the 5-fold CV was consistent across all the ML Models.

B. Topological Dataset (Table 5B)

We extracted six centrality/topological features from the mass flow graph and used it to train the ML models. We saw that there was improvement in the predictive power across the various ML models compared to adjacency features. Random Forest outperformed other models with an AuROC of 0.69 and Accuracy of 0.82. All models achieved relatively improved Accuracy and F1-Score values, indicating good overall performance. The accuracy in 5-fold cross-validation varies slightly across models but generally falls in the range of 0.74 to 0.85.

C. ReFeX Feature Set (Table 5C)

We extracted 31 feature sets from mass flow graph using GraphRole. When trained with the ML models, we observed, all models improved performance on this feature sets. Random Forest had the highest AuROC (0.69) and Accuracy (0.85) among all models, indicating its effectiveness with ReFeX features. Logistic Regression and Decision Tree also showed reasonable performance with AuROC values around 0.62 and Accuracy above 0.80. Naive Bayes had a relatively low performance with an AuROC of 0.63 and an Accuracy of 0.58.

D. RolX Feature Set (Table 5D)

We generated a set of 5 features based on role clustering of the ReFeX features with GraphRole Algorithm and trained ML models on them. Naive Bayes achieved the highest AuROC (0.56), Accuracy (0.79), and F1-Score (0.73). Random Forest and SVM also showed reasonable performance with AuROC values of 0.55 and 0.56, respectively. The accuracy in 5-fold cross-validation is relatively consistent for all models, with scores mostly around 0.74 to 0.79.

E. ReFeX and RolX Feature Set (Table 5E)

We combined ReFeX and RolX features set to form a new dataset which we used for ML training. Random Forest achieved the highest AuROC (0.68) and Accuracy (0.83) among all models, indicating its effectiveness in utilizing the combined features and performing well on the dataset. Logistic Regression, Decision Tree, and KNN also showed reasonable performance with AuROC values around 0.66, 0.62, and 0.59, respectively. Naive Bayes had the lowest Accuracy (0.61) among all models. While it achieved a relatively high Precision (0.75), its Recall (0.61) and F1-Score (0.64) were lower, indicating that it may struggle to correctly identify positive instances.

F. TopologyReFeX Feature Set (Table 5F)

We combined Topological and ReFeX feature sets to form a new dataset (TopologyReFeX). Random Forest achieved the highest AuROC (0.69) and Accuracy (0.85) among all models, indicating its strong performance on the dataset with TopologyReFeX features. KNN also performed well with an AuROC of 0.62 and an Accuracy of 0.82, demonstrating its capability to handle the data effectively. Logistic Regression, Decision Tree, and Naive Bayes showed moderate performance, with AuROC values between 0.62 and 0.63 and Accuracies ranging from 0.58 to 0.79. SVM had the lowest AuROC (0.53) and Accuracy (0.79) compared to other models on this dataset.

G. TopologyRolX Feature Set (Table 5G)

We combined Topological and RolX datasets to form a new dataset as TopologyRolX. KNN achieved the highest AuROC (0.66) and Accuracy (0.80) among all models, indicating its strong performance on the dataset with TopologyRolX features. Random

Forest and Decision Tree also showed competitive performance with AuROC values around 0.67 and Accuracies of 0.82. Logistic Regression, SVM, and Naive Bayes had relatively lower performance, with AuROC values between 0.46 and 0.52 and Accuracies ranging from 0.67 to 0.77.

H. TopologyReFeXRolX Dataset (Table 5H)

We finally combined the three-feature set, Topological, ReFeX and RolX features to form a new dataset. Random Forest achieved the highest AuROC (0.66) and Accuracy (0.83) among all models, indicating its strong performance on the dataset with TopologyReFexRolX features. Logistic Regression, Decision Tree, and KNN also showed competitive performance with AuROC values around 0.61 to 0.63 and Accuracies ranging from 0.76 to 0.80. Naive Bayes had a higher AuROC (0.65) compared to other datasets, but it showed the lowest Accuracy (0.61) and F1-Score (0.64). SVM had the lowest AuROC (0.53) and performed relatively lower in terms of Accuracy (0.79) and F1-Score (0.71) on this dataset.

Overall, Random Forest classifier with 500 trees and a maximum depth of 42 was found to provide the best results across various feature sets and overall best in ReFeX feature set after the experiment. Information gain was the criteria used to determine the best tree splits, and all feature sets except Adjacency and RolX utilized log2 (2k) features. As shown in Table 4.2, the model's test set assessment demonstrated an overall accuracy of 85%. The model also showed an 85% accuracy rate and an 83% recall rate when the ReFeX and Combined Topological&ReFeX, and Topological&ReFeX&RolX feature sets were considered. However, we realize ReFeX has a significant impact on how RF performs on each of the combinations that contain the ReFeX feature set. A heatmap shown in Figure 4.2 shows the record of the Accuracy of the ML models across the different datasets.

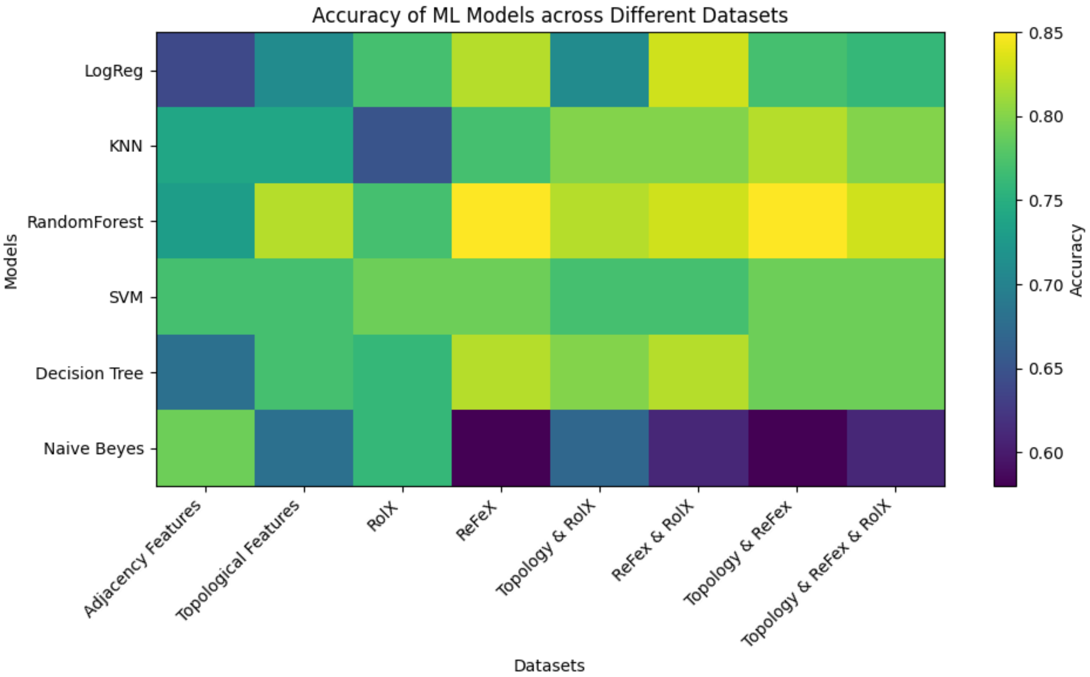


Figure 4.2: Heatmap of Accuracy of ML models across different dataset

Table 4.2: Performance evaluation for Random Forest over eight datasets

Random Forest					
Dataset	AuROC	Accuracy	Precision	Recall	F1-score
Adjacency	0.52	0.73	0.67	0.73	0.69
Topological	0.69	0.82	0.8	0.82	0.81
ReFeX	**0.69**	**0.85**	**0.85**	**0.85**	**0.83**
RolX	0.55	0.77	0.72	0.77	0.72
ReFeX and RolX	0.68	0.83	0.82	0.83	0.81
Topological and RolX	0.67	0.82	0.8	0.82	0.8
Topological and ReFeX	**0.69**	**0.85**	**0.85**	**0.85**	**0.83**
TopologicalReFeXRolX	**0.69**	**0.85**	**0.85**	**0.85**	**0.83**

Table 4.2 shows performance matrix of Random Forest (RF) on the various feature sets, Adjacency, ReFeX, RolX, Topological Features and the combinations of ReFeX&RolX, Topology&ReFeX, Topology&RolX and Topological&ReFeX&RolX. The Model was trained on 80% of the nodes of the MFG and the reported performance metrics were computed on a held-out dataset with 20% of nodes. A heatmap in Figure 4.3 shows the performance of RF model across various datasets and performance evaluation metrics.

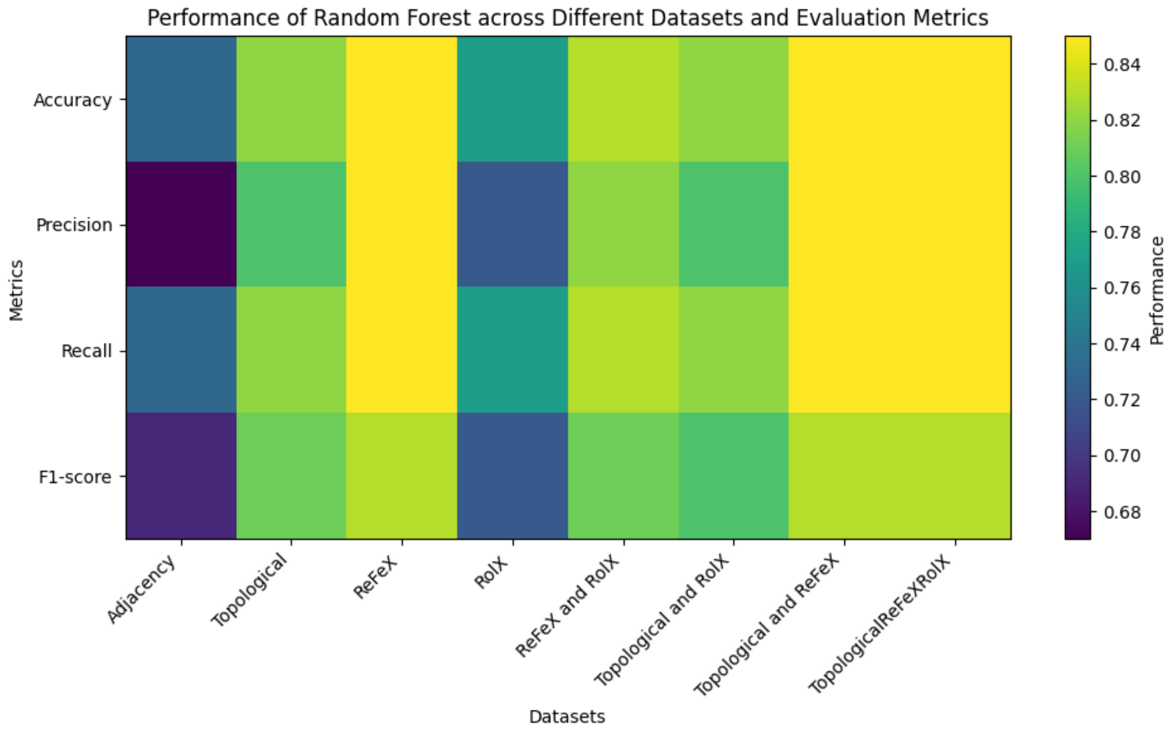


Figure 4.3: The Heatmap representing the Performance of RF across Different datasets and evaluation metrics.

We examined the confusion matrix (Figure 4.4A), and it suggested that the classifier is relatively bad at predicting the non-essential reactions (with an accuracy of 40%), but it shows near state-of-the-art accuracy for essential genes (with an accuracy of 98.04%). This discrepancy could be explained by the fact that the non-essential reactions are not as well represented in the dataset as the essential reactions are.

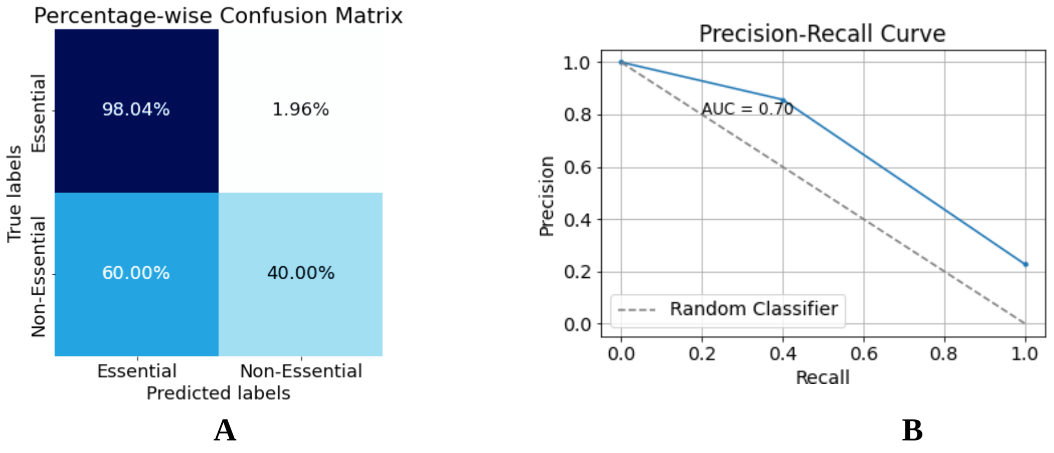


Figure 4.4: Gene essentiality prediction in *Plasmodium Falciparum* (iAM_Pf480)

(A) Confusion Matrix of Random Forest on ReFeX Features and (B) Precision-Recall Curve of Random Forest. We see AUC = 0.7 indicates that the model's ability to differentiate between the positive and negative classes is of moderate strength.

The thorough classification report shown in Table 4.3 shows that for ReFeX features, the model obtains equivalent accuracy values for both essential and non-essential classes. The Recall and F-Score for essential reactions, however, are much higher than those for non-essential reactions. The macro or class average, which reflects the unweighted average of precision, recall, and F-Score for each class, is lower than the weighted average in recall and F1-Score but produces the same precision score. This suggests that although the model failed to accurately identify certain non-essential reactions, it fared noticeably better on those essential reactions. This resulted in lower recall and F1 scores for that class of reactions. The number of samples in each class is taken into consideration by the weighted average, which results in a more balanced evaluation of overall performance.

Table 4.3: Detail Classification Report of RF on ReFeX features on the test dataset.

Random Forest (Binary Classifier)				
	Precision	Recall	f1-Score	Support
Essential	0.85	0.98	0.91	51
Non-Essential	0.86	0.4	0.55	15
Accuracy			**0.85**	66
Macro Avg	0.85	0.69	0.73	66
Weighted Avg	0.85	0.85	0.83	66

AuROC of (A). Naive Bayes (NB), (B). Decision Tree (DC), (C). Support Vector Machine (SVM), (D). k-nearest neighbour (kNN), (E). Logistic regression (LogReg) and (F). Random Forest (RF). We observed high AuROC scores for ReFeX features or Any of its combinations. RF records highest AuROC of 0.69 in Topological Features set, ReFeX feature set, combination of Topological Feature set and combination of Topological & ReFeX & RolX but will record lower AuROC in Adjacency Features. A heatmap plot of AuROC of the various ML models across all datasets is shown in Figure 4.5. These findings suggest that the Random Forest model performed particularly well in identifying essential and nonessential genes when using the ReFeX characteristics.

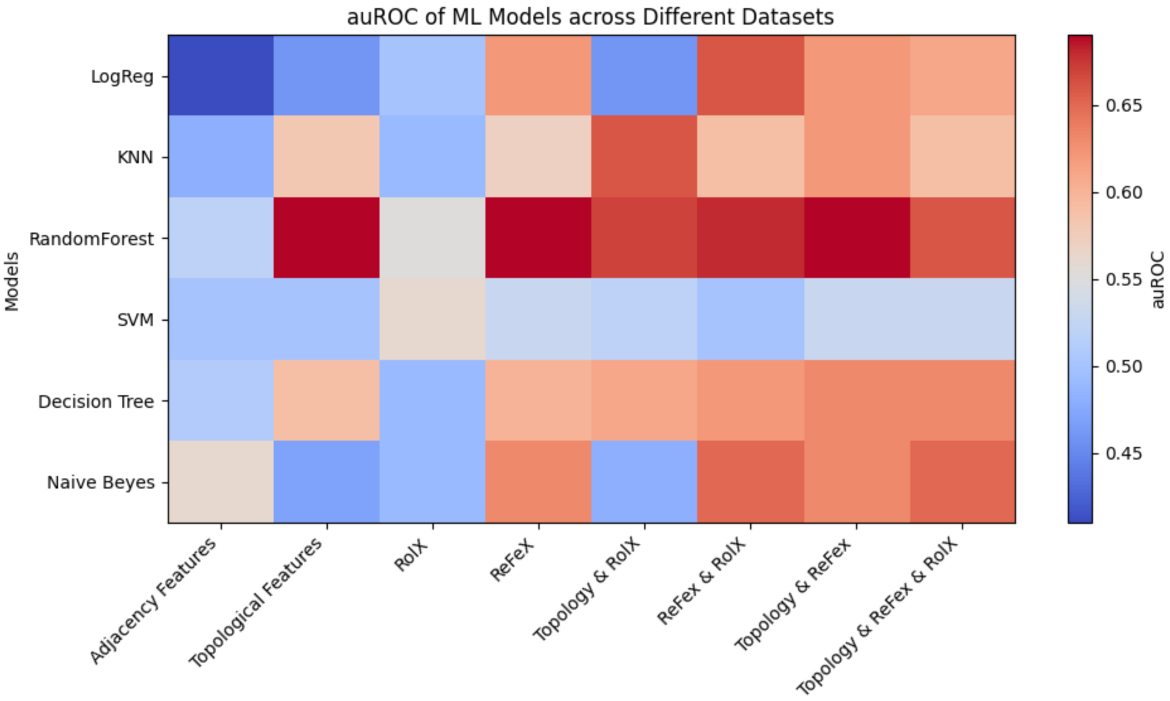


Figure 4.5: Plot of Area Under the Receiver Operating Characteristic (AuROC) of Binary Classifiers over the various feature sets.

4.4 Comparative Analysis with FBA

In this work, we used the COBRApy library tool to conduct Single Reaction Deletion analysis on the Genome-Scale Metabolic (GSM) model of *Plasmodium falciparum*. This study' primary goal was to determine how well Flux Balance study (FBA) could predict the set of reactions that were noted on the Flux-weighted reaction centric graph.

In a confusion matrix, which offers a thorough breakdown of the performance of FBA predictions compared to the actual labels of the reactions in the dataset, we reported the findings of this investigation. True positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are the four categories that the confusion matrix displays (Table 4.4).

Table 4.4: Confusion Matrix on FBA Predictions on the dataset

		Predicted	
		Essential	Non-Essential
True Labels	Essential	138	120
	Non-Essential	17	55
	Accuracy		**0.59**

According to the stated accuracy of FBA predictions, which was 0.58, 138 out of the 258 essential reactions in the dataset were accurately recognised by FBA as being essential (TP). But it also mislabelled 55 non-essential reactions as essential (FP) and 120 essential reactions as non-essential (FN). When predicting the essentiality of responses, the machine learning (ML) model used in this research performed better, with an accuracy of 0.85. This indicates that the ML model fared better than the FBA model in terms of accuracy, since more accurate predictions were created on the dataset.

The findings show that while FBA may provide some insight into the need for responses, it may not be as precise as the ML model used in this work. The ML model's improved accuracy suggests that it has the ability to more accurately classify and predict essential genes, which is critical for comprehending the metabolic behaviour of the organism under investigation. Overall, the results imply that integrating both FBA and ML techniques might provide a more thorough and accurate examination of metabolic essentiality, assisting researchers in gaining deeper insights into how the metabolic network in the organism functions.

4.5 Interesting Biological Findings

Comparing the ML model prediction to the label, we identified 9 genes that were labelled nonessential but predicted as essential (False Positives) and hence decided to search in literature and acquire more insight into these genes, seeking to find experimental evidence of its essentiality in literature and these genes have been listed in Table 4.5.

Table 4.5: List of False Positive Prediction (Genes Labeled as non-essential but predicted as essential)

Reaction	Gene	Binary labels	ML Prediction
ACONTb	PF3D7_1342100	NE	E
MAN6PI	PF3D7_0801800	NE	E
PPPGO6m	PF3D7_1028100	NE	E
PYNP2r	PF3D7_0513300	NE	E
TMPPP	PF3D7_0614000	NE	E
THBPT4ACAMDASE	PF3D7_1108300	NE	E
CITtcm	PF3D7_1223800	NE	E
DHORTS	PF3D7_1472900	NE	E
SUCOAS1m	(PF3D7_1437700 or PF3D7_1431600) and PF3D7_1108500	NE	E

We present a discussion of experimental evidence found in the literature regarding specific genes and their potential applications in malaria drug discovery:

1. Gene PF3D7_1342100 encodes for Aconitase, an enzyme responsible for catalyzing the stereo-specific isomerization of citrate to isocitrate via cis-aconitate in the tricarboxylic acid cycle. A study conducted by Ke *et al.* in 2015 revealed that this gene plays a crucial role in the Tricarboxylic Acid Cycle in the mitochondrion of *Plasmodium falciparum*. Knocking out this gene resulted in the parasite's inability to fully utilize glucose nutrients in the TCA cycle, affecting its carbon source. As a consequence, the parasite could not mature into gametocytes, hindering gamete formation. This study provides valuable experimental evidence to investigate further (Ke *et al., 2*015).
2. Gene PF3D7_0801800 codes for mannose-6-phosphate isomerase, which is currently under investigation in *Plasmodium* berghei, a pathogen responsible for cerebral malaria. Lv *et al.* (2022) found that administering D-mannose to *Plasmodium* berghei-infected mice resulted in weight loss and reduced parasitemia without noticeable side effects, suggesting a potential role of this gene in malaria pathogenesis (Lv *et al., 2*022).

3. Gene PF3D7_1028100 encodes for protoporphyrinogen oxidase (PfPPO), localized in the mitochondria and active under anaerobic conditions. PfPPO depends on electron transport chain (ETC) acceptors for its activity. Notably, ETC inhibitors, such as atovaquone and antimycin, inhibit the enzyme's function. Atovaquone, a known parasite dihydroorotate dehydrogenase inhibitor, inhibits heme synthesis in *P. falciparum* culture and has been used to design Atovaquone-proguanil, an antimalarial drug (Nagaraj *et al., 2*010; Nixon *et al., 2*013)
4. Gene PF3D7_0513300 encodes for purine nucleoside phosphorylase (PfPNP), representing a potential target for antimalarial drug design. Inhibition of PfPNP has been shown to effectively kill malaria parasites both in vitro and in vivo (Dziekan *et al., 2*019). However, currently known inhibitors, immucillins, are orally available and exhibit low toxicity to animals and humans. Yet, none of these compounds have entered clinical trials for malaria treatment (Holanda *et al., 2*020; Kagami *et al., 2*017).
5. For the remaining genes (PF3D7_0614000, PF3D7_1108300, PF3D7_1223800, PF3D7_1472900, PF3D7_1437700 (or PF3D7_1431600), and PF3D7_1108500), there is no literature evidence suggesting their direct biological relevance in malaria drug discovery. Further research is required to gain insight into their potential roles in the malaria parasite's metabolism and pathogenesis.

CHAPTER FIVE
CONCLUSION AND RECOMMENDATIONS

5.1 Summary

This research focused on the challenging task of predicting metabolic important genes in eukaryotes, with a specific focus on the pathogenic organism *Plasmodium falciparum*, which causes malaria. Previous studies in this area primarily dealt with prokaryotes, and their methods often failed to adequately represent the weighted, directed nature of metabolite transport in metabolic networks (Freischem *et al.*, 2022; Kim *et al.*, 2019).

To address this issue, we designed a Network-based Machine Learning framework that explored various network properties in *Plasmodium falciparum* using the Genome-Scale Metabolic Model (iAM_Pf480) collected from the BiGG database and essentiality data from the Ogee Database. By considering the direct weighted structure of the metabolic network and employing advanced network-based features, the machine learning framework achieved a significant improvement in the accuracy of gene essentiality, achieving state-of-the-art results in the study of metabolic genes in *Plasmodium falciparum* eukaryote.

5.2 Contribution to knowledge

This study has enhanced our understanding of the complexity of metabolic networks and their role in determining gene essentiality. Notably, we identified key genes labelled as non-essential in the Ogee database but were predicted as essential by our model. Many of these genes had been previously identified as potential drug targets for malaria treatment, suggesting promising avenues for further investigation.

5.3 Limitations and Recommendation

The limitation of this study lies in its focus on *Plasmodium falciparum* alone which prevents it generalizability, hence, necessitating further exploration of this approach in other eukaryotic pathogens. Additionally, the quality of Genome-Scale Metabolic Models significantly influences metabolic essentiality predictions and should be considered in future research.

REFERENCES

Abdel-Haleem, Alyaa M., Hooman Hefzi, Katsuhiko Mineta, Xin Gao, Takashi Gojobori, Bernhard O. Palsson, Nathan E. Lewis, and Neema Jamshidi. 2018. "Functional Interrogation of *Plasmodium* Genus Metabolism Identifies Species- and Stage-Specific Differences in Nutrient Essentiality and Drug Targeting." *PLOS Computational Biology* 14(1):e1005895. doi: 10.1371/journal.pcbi.1005895.

Acencio, Marcio L., and Ney Lemke. 2009. "Towards the Prediction of Essential Genes by Integration of Network Topology, Cellular Localization and Biological Process Information." *BMC Bioinformatics* 10(1).

Aleryani, Aliya, Wenjia Wang, and Beatriz de la Iglesia. 2020. "Multiple Imputation Ensembles (MIE) for Dealing with Missing Data." *SN Computer Science* 1(3).

Aromolaran, Olufemi, Damilare Aromolaran, Itunuoluwa Isewon, and Jelili Oyelade. 2021. "Machine Learning Approach to Gene Essentiality Prediction: A Review." *Briefings in Bioinformatics* 22(5). doi: 10.1093/bib/bbab128.

Aromolaran, Olufemi, Thomas Beder, Eunice Adedeji, Yvonne Ajamma, Jelili Oyelade, Ezekiel Adebiyi, and Rainer Koenig. 2021. "Predicting Host Dependency Factors of Pathogens in Drosophila Melanogaster Using Machine Learning." *Computational and Structural Biotechnology Journal* 19:4581–92.

Aromolaran, Olufemi, Thomas Beder, Marcus Oswald, Jelili Oyelade, Ezekiel Adebiyi, and Rainer Koenig. 2020. "Essential Gene Prediction in Drosophila Melanogaster Using Machine Learning Approaches Based on Sequence and Functional Features." *Computational and Structural Biotechnology Journal* 18:612–21. doi: https://doi.org/10.1016/j.csbj.2020.02.022.

Ashtiani, Minoo, Ali Salehzadeh-Yazdi, Zahra Razaghi-Moghadam, Holger Hennig, Olaf Wolkenhauer, Mehdi Mirzaie, and Mohieddin Jafari. 2018. "A Systematic Survey of Centrality Measures for Protein-Protein Interaction Networks." *BMC Systems Biology* 12(1):1–17.

Ashtiani, Salehzadeh-Yazdi, Razaghi-Moghadam, Hennig, Wolkenhauer, Mirzaie, and Jafari. 2018. "A Systematic Survey of Centrality Measures for Protein-Protein Interaction Networks." *BMC Systems Biology* 12(1):1–17. doi: 10.1186/s12918-018-0598-2.

Azhagesan, Karthik, Balaraman Ravindran, and Karthik Raman. 2018. "Network-Based Features Enable Prediction of Essential Genes across Diverse Organisms." *PLOS ONE* 13(12):e0208722.

Bandoim, Lana. 2019. “Cellular Metabolism: Definition, Process & the Role of AT*P*.”

Becker, Scott A., Adam M. Feist, Monica L. Mo, Gregory Hannum, Bernhard Ø. Palsson, and Markus J. Herrgard. 2007. “Quantitative Prediction of Cellular Metabolism with Constraint-Based Models: The COBRA Toolbox.” *Nature Protocols* 2(3):727–38. doi: 10.1038/nprot.2007.99.

Beder, Thomas, Olufemi Aromolaran, J\" urgen D\" onitz, Sofia Tapanelli, Eunice O. Adedeji, Ezekiel Adebiyi, Gregor Bucher, and Rainer Koenig. 2021. “Identifying Essential Genes across Eukaryotes by Machine Learning.” *NAR Genomics and Bioinformatics* 3(4).

Beguerisse-Díaz, Mariano, Gabriel Bosque, Diego Oyarzún, Jesús Picó, and Mauricio Barahona. 2018. “Flux-Dependent Graphs for Metabolic Networks.” *Npj Systems Biology and Applications* 4(1). doi: 10.1038/s41540-018-0067-y.

Bernstein, David B., Snorre Sulheim, Eivind Almaas, and Daniel Segr\` e. 2021. “Addressing Uncertainty in Genome-Scale Metabolic Model Reconstruction and Analysis.” *Genome Biology* 22(1).

Bordbar, Aarash, Jonathan M. Monk, Zachary A. King, and Bernhard O. Palsson. 2014. “Constraint-Based Models Predict Metabolic and Associated Cellular Functions.” *Nature Reviews Genetics* 15(2):107–20.

Brahmaih, Kala. 2020. “Graph Mining and Exploration Techniques.”

Buchweitz, Lea F., James T. Yurkovich, Christoph Blessing, Veronika Kohler, Fabian Schwarzkopf, Zachary A. King, Laurence Yang, Freyr J\’ ohannsson, \’ Olafur E Sigurj\’ onsson, \’ Ottar Rolfsson, Julian Heinrich, and Andreas Dr\" ager. 2020. “Visualizing Metabolic Network Dynamics through Time-Series Metabolomic Data.” *BMC Bioinformatics* 21(1).

Cakmak, Ali, and M. Hasan Celik. 2021. “Personalized Metabolic Analysis of Diseases.” *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 18(3):1014–25.

Camacho, Diogo M., Katherine M. Collins, Rani K. Powers, James C. Costello, and James J. Collins. 2018. “Next-Generation Machine Learning for Biological Networks.” *Cell* 173(7):1581–92. doi: 10.1016/j.cell.2018.05.015.

Campos, Tulio L., Pasi K. Korhonen, Andreas Hofmann, Robin B. Gasser, and Neil D. Young. 2022. “Harnessing Model Organism Genomics to Underpin the Machine Learning-Based Prediction of Essential Genes in Eukaryotes – Biotechnological Implications.” *Biotechnology Advances* 54:107822. doi: https://doi.org/10.1016/j.biotechadv.2021.107822.

Carey, Maureen A., Jason A. Papin, and Jennifer L. Guler. 2017. “Novel *Plasmodium falciparum* Metabolic Network Reconstruction Identifies Shifts Associated with Clinical Antimalarial Resistance.” *BMC Genomics* 18(1). doi: 10.1186/s12864-017-3905-1.

Centers for Disease Control (CDC). 2021. “CDC - Malaria - Malaria Worldwide.”

Cheng, Jian, Wenwu Wu, Yinwen Zhang, Xiangchen Li, Xiaoqian Jiang, Gehong Wei, and Shiheng Tao. 2013. “A New Computational Strategy for Predicting Essential Genes.” *BMC Genomics* 14(1). doi: 10.1186/1471-2164-14-910.

Chiappino-Pepe, Anush, Vikash Pandey, and Oliver Billker. 2021. “Genome Reconstructions of Metabolism of *Plasmodium* RBC and Liver Stages.” *Current Opinion in Microbiology* 63:259–66. doi: 10.1016/j.mib.2021.08.006.

Christiansen, Kylie N. 2022. *Analysis of Protein-Protein Interaction Networks in Aging Flight Muscle of the Male Hawk Moth, Manduca Sexta*.

Cover, T. M., and *P*. E. Hart. 1967. “Nearest Neighbor Pattern Classification.” *IEEE Transactions on Information Theory* 13(1):21–27. doi: 10.1109/TIT.1967.1053964.

Cuevas, Daniel A., Janaka Edirisinghe, Chris S. Henry, Ross Overbeek, Taylor G. O’Connell, and Robert A. Edwards. 2016. “From DNA to FBA: How to Build Your Own Genome-Scale Metabolic Model.” *Frontiers in Microbiology* 7.

Cuperlovic-Culf, Miroslava. 2018. “Machine Learning Methods for Analysis of Metabolic Data and Metabolic Pathway Modeling.” *Metabolites* 8(1):4. doi: 10.3390/metabo8010004.

Donges, Niklas. 2021. “Random Forest: A Complete Guide for Machine Learning.”

Dusad, Varshit, Denise Thiel, Mauricio Barahona, Hector C. Keun, and Diego A. Oyarzún. 2021. “Opportunities at the Interface of Network Science and Metabolic Modeling.” *Frontiers in Bioengineering and Biotechnology* 8. doi: 10.3389/fbioe.2020.591049.

Dziekan, Jerzy M., Han Yu, Dan Chen, Lingyun Dai, Grennady Wirjanata, Andreas Larsson, Nayana Prabhu, Radoslaw M. Sobota, Zbynek Bozdech, and P\" ar Nordlund. 2019. “Identifying Purine Nucleoside Phosphorylase as the Target of Quinine Using Cellular Thermal Shift Assay.” *Science Translational Medicine* 11(473).

Ebrahim, Ali, Joshua A. Lerman, Bernhard O. Palsson, and Daniel R. Hyduke. 2013. “COBRApy: COnstraints-Based Reconstruction and Analysis for Python.” *BMC Systems Biology* 7(1). doi: 10.1186/1752-0509-7-74.

Erciyes, K. 2015. “Analysis of Biological Networks.” Pp. 213–40 in *Computational Biology*. Cham: Springer International Publishing.

Ferreira, António E. N., Marta Sousa Silva, and Carlos Cordeiro. 2021. “Metabolic Network Inference from Time Series.” Pp. 127–33 in *Systems Medicine*. Elsevier.

Freischem, Lilli J., Mauricio Barahona, and Diego A. Oyarzún. 2022. *Prediction of Gene Essentiality Using Machine Learning and Genome-Scale Metabolic Models*. Cold Spring Harbor Laboratory.

Gatto, Francesco, Heike Miess, Almut Schulze, and Jens Nielsen. 2015. “Flux Balance Analysis Predicts Essential Genes in Clear Cell Renal Cell Carcinoma Metabolism.” *Scientific Reports* 5(1). doi: 10.1038/srep10738.

Gurobi. 2020. “The Leader in Decision Intelligence Technology - Gurobi Optimization.” Retrieved (http://www.gurobi.com/).

Hameri, Tuure, Georgios Fengos, Meric Ataman, Ljubisa Miskovic, and Vassily Hatzimanikatis. 2019. “Kinetic Models of Metabolism That Consider Alternative Steady-State Solutions of Intracellular Fluxes and Concentrations.” *Metabolic Engineering* 52:29–41. doi: 10.1016/j.ymben.2018.10.005.

Hasan, Md Abid, and Stefano Lonardi. 2020. “DeeplyEssential: A Deep Neural Network for Predicting Essential Genes in Microbes.” *BMC Bioinformatics* 21(S14). doi: 10.1186/s12859-020-03688-y.

Henderson, Keith, Brian Gallagher, Tina Eliassi-Rad, Hanghang Tong, Sugato Basu, Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li. 2012. “RolX.” ACM.

Henderson, Keith, Brian Gallagher, Lei Li, Leman Akoglu, Tina Eliassi-Rad, Hanghang Tong, and Christos Faloutsos. 2011. “It’s Who You Know.” ACM.

Holanda, Rudson J., Candida Deves, Leandro S. Moreira-Dill, Cesar L. Guimar∼ aes, Leonardo K. B. Marttinelli, Carla F. C. Fernandes, Patr\’ icia S. M. Medeiros, Soraya S. Pereira, Eduardo R. Honda, Rodrigo G. St\’ abeli, Di\’ ogenes S. Santos, Andreimar M. Soares, and Luiz H. Pereira da Silva. 2020. “*Plasmodium falciparum* Purine Nucleoside Phosphorylase as a Model in the Search for New Inhibitors by High Throughput Screening.” *International Journal of Biological Macromolecules* 165:1832–41.

Hua, Hong-Li, Fa-Zhan Zhang, Abraham Alemayehu Labena, Chuan Dong, Yan-Ting Jin, and Feng-Biao Guo. 2016. “An Approach for Predicting Essential Genes Using Multiple Homology Mapping and Machine Learning Algorithms.” *BioMed Research International* 2016:1–9. doi: 10.1155/2016/7639397.

Iranzadeh, Arash, and Nicola Jane Mulder. 2019. “Bacterial Pan-Genomics.” Pp. 21–38 in *Microbial Genomics in Sustainable Agroecosystems*. Singapore: Springer Singapore.

Jeong, H., S. *P*. Mason, A. L. Barab\’ asi, and Z. N. Oltvai. 2001. “Lethality and Centrality in Protein Networks.” *Nature* 411(6833):41–42.

Jone, Dixon. 2023. “Google PageRank Explained for SEO Beginners.” *Search Engine Journal*. Retrieved July 23, 2023 (https://www.searchenginejournal.com/google-pagerank/483521/).

Kagami, Luciano Porto, Gustavo Machado das Neves, Ricardo Pereira Rodrigues, Vinicius Barreto da Silva, Vera Lucia Eifler-Lima, and Daniel F\’ abio Kawano. 2017. “Identification of a Novel Putative Inhibitor of the *Plasmodium falciparum* Purine Nucleoside Phosphorylase: Exploring the Purine Salvage Pathway to Design New Antimalarial Drugs.” *Molecular Diversity* 21(3):677–95.

Kanti Kumar, Pijush, Anurag Dutta, and Padmanavan Kumar. 2023. “Application of Graph Mining Algorithms for the Analysis of Web Data.” *SSRN Electronic Journal*.

Kaslovsky, Daniel. 2019. “GitHub - Dkaslovsky/GraphRole: Automatic Feature Extraction and Node Role Assignment for Transfer Learning on Graphs (ReFeX & RolX).”

Ke, Hangjun, Ian A. Lewis, Joanne M. Morrisey, Kyle J. McLean, Suresh M. Ganesan, Heather J. Painter, Michael W. Mather, Marcelo Jacobs-Lorena, Manuel Llin\’ as, and Akhil B. Vaidya. 2015. “Genetic Investigation of Tricarboxylic Acid Metabolism during the *Plasmodium falciparum* Life Cycle.” *Cell Reports* 11(1):164–74.

Kim, Eun-Youn, Daniel Ashlock, and Sung Ho Yoon. 2019. “Identification of Critical Connectors in the Directed Reaction-Centric Graphs of Microbial Metabolic Networks.” *BMC Bioinformatics* 20(1).

Kumar, D. Thirumal, *P*. Sneha, Jennifer Uppin, S. Usha, and C. George Priya Doss. 2018. “Chapter Eight - Investigating the Influence of Hotspot Mutations in Protein–Protein Interaction of IDH1 Homodimer Protein: A Computational Approach.” Pp. 243–61 in *Protein-Protein Interactions in Human Disease, Part B*. Vol. 111, *Advances in Protein Chemistry and Structural Biology*, edited by R. Donev. Academic Press.

Li, min Li, Hao Jiang, Yushan Qiu, Wai-Ki Ching, and Vassilios S. Vassiliadis. 2013. “Discovery of Metabolite Biomarkers: Flux Analysis and Reaction-Reaction Network Approach.” *BMC Systems Biology* 7(2):1–7.

Li, Xingyi, Wenkai Li, Min Zeng, Ruiqing Zheng, and Min Li. 2019. “Network-Based Methods for Predicting Essential Genes or Proteins: A Survey.” *Briefings in Bioinformatics* 21(2):566–83. doi: 10.1093/bib/bbz017.

Loem, Mengsay. 2021. “What Is Network Analysis?” *Towards Data Science*.

Lv, Li, Zihao Xu, Meichen Zhao, Jian Gao, Rumeng Jiang, Qian Wang, and Xiaoyu Shi. 2022. “Mannose Inhibits *Plasmodium* Parasite Growth and Cerebral Malaria

Development via Regulation of Host Immune Responses." *Frontiers in Immunology* 13.

Machicao, Jeaneth, Francesco Craighero, Davide Maspero, Fabrizio Angaroni, Chiara Damiani, Alex Graudenzi, Marco Antoniotti, and Odemir M. Bruno. 2021. "On the Use of Topological Features of Metabolic Networks for the Classification of Cancer Samples." *Current Genomics* 22(2):88–97. doi: 10.2174/1389202922666210301084151.

Martins Conde, Patricia do Rosario, Thomas Sauter, and Thomas Pfau. 2016. "Constraint Based Modeling Going Multicellular." *Frontiers in Molecular Biosciences* 3.

Masutin, Viktor, Christian Kersch, and Simone SchmitzSpanke. 2022. "A Systematic Review: Metabolomicsbased Identification of Altered Metabolites and Pathways in the Skin Caused by Internal and External Factors." *Experimental Dermatology* 31(5):700–714.

Medlock, Gregory L., and Jason A. Papin. 2020. "Guiding the Refinement of Biochemical Knowledgebases with Ensembles of Metabolic Networks and Machine Learning." *Cell Systems* 10(1):109-119.e3.

Milano, Marianna, Giuseppe Agapito, and Mario Cannataro. 2022. "Challenges and Limitations of Biological Network Analysis." *BioTech* 11(3):24.

Muzio, Giulia, Leslie O'Bray, and Karsten Borgwardt. 2020. "Biological Network Analysis with Deep Learning." *Briefings in Bioinformatics* 22(2):1515–30. doi: 10.1093/bib/bbaa257.

Naderi Yeganeh, Pourya, Chrsitine Richardson, Erik Saule, Ann Loraine, and M. Taghi Mostafavi. 2020. "Revisiting the Use of Graph Centrality Models in Biological Pathway Analysis." *BioData Mining* 13(1).

Nagaraj, Viswanathan Arun, Rajavel Arumugam, Dasari Prasad, Pundi N. Rangarajan, and Govindarajan Padmanaban. 2010. "Protoporphyrinogen IX Oxidase from *Plasmodium falciparum* Is Anaerobic and Is Localized to the Mitochondrion." *Molecular and Biochemical Parasitology* 174(1):44–52.

Nandi, Sutanu, Piyali Ganguli, and Ram Rup Sarkar. 2020. "Essential Gene Prediction Using Limited Gene Essentiality Information–An Integrative Semi-Supervised Machine Learning Strategy." *PLOS ONE* 15(11):e0242943. doi: 10.1371/journal.pone.0242943.

Nandi, Sutanu, Abhishek Subramanian, and Ram Rup Sarkar. 2017. "An Integrative Machine Learning Strategy for Improved Prediction of Essential Genes in Escherichia

Coli Metabolism Using Flux-Coupled Features.” *Molecular BioSystems* 13(8):1584–96. doi: 10.1039/c7mb00234c.

Nigatu, Dawit, Patrick Sobetzko, Malik Yousef, and Werner Henkel. 2017. “Sequence-Based Information-Theoretic Features for Gene Essentiality Prediction.” *BMC Bioinformatics* 18(1). doi: 10.1186/s12859-017-1884-5.

Nixon, G. L., D. M. Moss, A. E. Shone, D. G. Lalloo, N. Fisher, *P*. M. O’Neill, S. A. Ward, and G. A. Biagini. 2013. “Antimalarial Pharmacology and Therapeutics of Atovaquone.” *Journal of Antimicrobial Chemotherapy* 68(5):977–85.

Oyelade, Jelili, Itunuoluwa Isewon, Olufemi Aromolaran, Efosa Uwoghiren, Titilope Dokunmu, Solomon Rotimi, Oluwadurotimi Aworunse, Olawole Obembe, and Ezekiel Adebiyi. 2019. “Computational Identification of Metabolic Pathways Of*Plasmodium falciparum*using Thek-Shortest Path Algorithm.” *International Journal of Genomics* 2019:1–13.

Oyelade, Jelili, Itunuoluwa Isewon, Efosa Uwoghiren, Olufemi Aromolaran, and Olufunke Oladipupo. 2018. “In Silico Knockout Screening of *Plasmodium falciparum* Reactions and Prediction of Novel Essential Reactions by Analysing the Metabolic Network.” *BioMed Research International* 2018:1–11.

Plaimas, Eils, and König. 2010. “Identifying Essential Genes in Bacterial Metabolic Networks with Machine Learning Methods.” *BMC Systems Biology* 4(1):1–16. doi: 10.1186/1752-0509-4-56.

Plaimas, Mallm, Oswald, Svara, Sourjik, Eils, and König. 2008. “Machine Learning Based Analyses on Metabolic Networks Supports High-Throughput Knockout Screens.” *BMC Systems Biology* 2(1):1–11. doi: 10.1186/1752-0509-2-67.

Plata, Germán, Tzu-Lin Hsiao, Kellen L. Olszewski, Manuel Llinás, and Dennis Vitkup. 2010. “Reconstruction and Flux-balance Analysis of the *Plasmodium falciparum* Metabolic Network.” *Molecular Systems Biology* 6(1):408. doi: 10.1038/msb.2010.60.

Raman, Karthik, Nandita Damaraju, and Govind Krishna Joshi. 2013. “The Organisational Structure of Protein Networks: Revisiting the Centrality–Lethality Hypothesis.” *Systems and Synthetic Biology* 8(1):73–81. doi: 10.1007/s11693-013-9123-5.

Rigoulet, M., C. L. Bouchez, *P*. Paumard, S. Ransac, S. Cuvellier, S. Duvezin-Caubet, J. *P*. Mazat, and A. Devin. 2020. “Cell Energy Metabolism: An Update.” *Biochimica et Biophysica Acta (BBA) - Bioenergetics* 1861(11):148276.

Sahu, Ankur, Mary-Ann Blätke, Jędrzej Jakub Szymański, and Nadine Töpfer. 2021. “Advances in Flux Balance Analysis by Integrating Machine Learning and

Mechanism-Based Models." *Computational and Structural Biotechnology Journal* 19:4626–40. doi: 10.1016/j.csbj.2021.08.004.

Schinn, SongMin, Carly Morrison, Wei Wei, Lin Zhang, and Nathan E. Lewis. 2021. "A Genomescale Metabolic Network Model and Machine Learning Predict Amino Acid Concentrations in Chinese Hamster Ovary Cell Cultures." *Biotechnology and Bioengineering* 118(5):2118–23.

Schonfeld, Ethan, Edward Vendrow, Joshua Vendrow, and Elan Schonfeld. 2021. "On the Relation of Gene Essentiality to Intron Structure: A Computational and Deep Learning Approach." *Life Science Alliance* 4(6):e202000951. doi: 10.26508/lsa.202000951.

Schonlau, Matthias, and Rosie Yuyan Zou. 2020. "The Random Forest Algorithm for Statistical Learning." *The Stata Journal: Promoting Communications on Statistics and Stata* 20(1):3–29.

Shen, Y., J. Liu, G. Estiu, B. Isin, Y. Y. Ahn, D. S. Lee, A. L. Barabási, V. Kapatral, O. Wiest, and Z. N. Oltvai. 2010. "Blueprint for Antimicrobial Hit Discovery Targeting Metabolic Networks." *Proceedings of the National Academy of Sciences* 107(3):1082–87. doi: 10.1073/pnas.0909181107.

Surendran, Praveen, Isobel D. Stewart, Victoria *P*. W. Au Yeung, Maik Pietzner, Johannes Raffler, Maria A. W\" orheide, Chen Li, Rebecca F. Smith, Laura B. L. Wittemans, Lorenzo Bomba, Cristina Menni, Jonas Zierer, Niccol\` o Rossi, Patricia A. Sheridan, Nicholas A. Watkins, Massimo Mangino, Pirro G. Hysi, Emanuele Di Angelantonio, Mario Falchi, Tim D. Spector, Nicole Soranzo, Gregory A. Michelotti, Wiebke Arlt, Luca A. Lotta, Spiros Denaxas, Harry Hemingway, Eric R. Gamazon, Joanna M. M. Howson, Angela M. Wood, John Danesh, Nicholas J. Wareham, Gabi Kastenm\" uller, Eric B. Fauman, Karsten Suhre, Adam S. Butterworth, and Claudia Langenberg. 2022. "Rare and Common Genetic Determinants of Metabolic Individuality and Their Effects on Human Health." *Nature Medicine* 28(11):2321–32.

Takigawa, Ichigaku, and Hiroshi Mamitsuka. 2013. "Graph Mining: Procedure, Application to Drug Discovery and Recent Advances." *Drug Discovery Today* 18(1–2):50–57.

Tiwari, Arvind Kumar. 2020. "Introduction to Machine Learning." Pp. 41–51 in *Deep Learning and Neural Networks*. IGI Global.

Toubiana, David, Rami Puzis, Lingling Wen, Noga Sikron, Assylay Kurmanbayeva, Aigerim Soltabayeva, Maria del Mar Rubio Wilhelmi, Nir Sade, Aaron Fait, Moshe Sagi, Eduardo Blumwald, and Yuval Elovici. 2019. "Combined Network Analysis and Machine Learning Allows the Prediction of Metabolic Pathways from Tomato Metabolomics Data." *Communications Biology* 2(1).

Vijayakumar, Supreeta, Pattanathu K. S. M. Rahman, and Claudio Angione. 2020. "A Hybrid Flux Balance Analysis and Machine Learning Pipeline Elucidates Metabolic Adaptation in Cyanobacteria." *IScience* 23(12):101818.

Voit, Eberhard O. 2017. "The Best Models of Metabolism." *Wiley Interdisciplinary Reviews. Systems Biology and Medicine* 9(6).

Wang, Mengyuan, Haiying Wang, and Huiru Zheng. 2022. "A Mini Review of Node Centrality Metrics in Biological Networks." *International Journal of Network Dynamics and Intelligence* 99–110.

World Health Organization (WHO). 2023. "Malaria." *World Health Organization: WHO*.

Wu, Stephen Gang, Yuxuan Wang, Wu Jiang, Tolutola Oyetunde, Ruilian Yao, Xuehong Zhang, Kazuyuki Shimizu, Yinjie J. Tang, and Forrest Sheng Bao. 2016. "Rapid Prediction of Bacterial Heterotrophic Fluxomics Using Machine Learning and Constraint Programming." *PLOS Computational Biology* 12(4):e1004838. doi: 10.1371/journal.pcbi.1004838.

Yasemi, Mohammadreza, and Mario Jolicoeur. 2021. "Modelling Cell Metabolism: A Review on Constraint-Based Steady-State and Kinetic Approaches." *Processes* 9(2):322. doi: 10.3390/pr9020322.

Yu, Yongming, Licai Yang, Zhiping Liu, and Chuansheng Zhu. 2017. "Gene Essentiality Prediction Based on Fractal Features and Machine Learning." *Molecular BioSystems* 13(3):577–84. doi: 10.1039/c6mb00806b.

Zampieri, Guido, Supreeta Vijayakumar, Elisabeth Yaneske, and Claudio Angione. 2019. "Machine and Deep Learning Meet Genome-Scale Metabolic Modeling." *PLOS Computational Biology* 15(7):e1007084. doi: 10.1371/journal.pcbi.1007084.

Zhang, Xue, Marcio Luis Acencio, and Ney Lemke. 2016. "Predicting Essential Genes and Proteins Based on Machine Learning and Network Topological Features: A Comprehensive Review." *Frontiers in Physiology* 7.

Appendix A: Gene/Reaction list resulting from our Flux-weighted reaction centric graph of Pf GSM model

Reaction	Gene	Ogee labels	ML Prediction	FBA
3DSPHR	PF3D7_0409500	E	E	E
3HAD100c	PF3D7_1323000	E	E	NE
3HAD120c	PF3D7_1323000	E	E	NE
3HAD140c	PF3D7_1323000	E	E	NE
3HAD160c	PF3D7_1323000	E	E	NE
3HAD40c	PF3D7_1323000	E	E	NE
3HAD60c	PF3D7_1323000	E	E	NE
3HAD80c	PF3D7_1323000	E	E	NE
3OAR100c	PF3D7_0922900	NE	NE	NE
3OAR120c	PF3D7_0922900	NE	NE	NE
3OAR140c	PF3D7_0922900	NE	NE	NE
3OAR160c	PF3D7_0922900	NE	NE	NE
3OAR40c	PF3D7_0922900	NE	NE	NE
3OAR60c	PF3D7_0922900	NE	NE	NE
3OAR80c	PF3D7_0922900	NE	NE	NE
3OAS100c	PF3D7_0626300	NE	NE	NE
3OAS120c	PF3D7_0626300	NE	NE	NE
3OAS140c	PF3D7_0626300	NE	NE	NE
3OAS160c	PF3D7_0626300	NE	NE	NE
3OAS60c	PF3D7_0626300	NE	NE	NE
3OAS80c	PF3D7_0626300	NE	NE	NE
ACCOAC	(PF3D7_1460000 or PF3D7_1469600) and PF3D7_1026900	E	E	NE
ACCOACc	(PF3D7_1460000 or PF3D7_1469600) and PF3D7_1026900	E	E	NE
ACCOAtm	PF3D7_1036800	NE	NE	NE
ACGAMPM	PF3D7_1130000	E	E	E
ACGPID_18_0_18_1	PF3D7_0911000 or PF3D7_0624700	E	E	E
ACOATAc	PF3D7_0211400	NE	NE	NE
ACONTa	PF3D7_1342100	NE	NE	NE
ACONTb	PF3D7_1342100	NE	E	NE
ADA	PF3D7_1029600	E	E	NE
ADEt	PF3D7_1347200	E	E	NE

ADK1	PF3D7_1008900 or PF3D7_0110900 or PF3D7_0305800 or PF3D7_0816900	E	E	NE
ADK3m	PF3D7_0415600	NE	NE	E
ADNt	PF3D7_1347200	E	E	NE
ADSL1r	PF3D7_0206700	E	E	NE
ADSS	PF3D7_1354500	E	E	NE
AGPAT1_16_0_16_0	PF3D7_1444300	E	E	E
AGPAT1_16_0_18_0	PF3D7_1444300	E	E	E
AGPAT1_16_0_18_1	PF3D7_1444300	E	E	E
AGPAT1_16_0_18_2	PF3D7_1444300	E	E	E
AGPAT1_18_0_18_0	PF3D7_1444300	E	E	E
AGPAT1_18_0_18_1	PF3D7_1444300	E	E	E
AGPAT1_18_1_18_1	PF3D7_1444300	E	E	E
AGPAT1_18_1_18_2	PF3D7_1444300	E	E	E
AGPAT1_18_2_18_2	PF3D7_1444300	E	E	E
AHCi	PF3D7_0520900	E	E	E
AKGCITtm	PF3D7_0823900 or PF3D7_1223800	E	E	NE
AKGMALtm	PF3D7_0823900	E	E	NE
AKGtm	PF3D7_0823900	E	E	NE
ALASm	PF3D7_1246100	E	E	E
ALAt5r	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	E	E	NE
AMETtm	PF3D7_1241600	E	E	E
ARGt5r	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	E	E	NE
ASNt5r	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	E	E	NE
ASPCT	PF3D7_1344800	E	E	E

ATPS4m	PF3D7_0217100 and PF3D7_0703600 and PF3D7_0705900 and PF3D7_1147700 and PF3D7_1005800 and PF3D7_1235700 and PF3D7_1310000 and PF3D7_1311300 and PF3D7_0715500	E	E	NE
ATPt	PF3D7_1004800 or PF3D7_1037300	E	E	NE
ATPtm2	PF3D7_1037300 or PF3D7_0108400 or PF3D7_1368700	E	E	NE
CDIPTr_16_0_16_0	PF3D7_1315600	E	E	E
CDIPTr_16_0_18_0	PF3D7_1315600	E	E	E
CDIPTr_16_0_18_1	PF3D7_1315600	E	E	E
CDIPTr_18_0_18_0	PF3D7_1315600	E	E	E
CDIPTr_18_0_18_1	PF3D7_1315600	E	E	E
CDIPTr_18_1_18_1	PF3D7_1315600	E	E	E
CDPMEKc	PF3D7_0503100	E	E	E
CDS_16_0_16_0	PF3D7_1409900	E	E	E
CDS_16_0_18_0	PF3D7_1409900	E	E	E
CDS_16_0_18_1	PF3D7_1409900	E	E	E
CDS_18_0_18_0	PF3D7_1409900	E	E	E
CDS_18_0_18_1	PF3D7_1409900	E	E	E
CDS_18_1_18_1	PF3D7_1409900	E	E	E
CEPTC_16_0_16_0	PF3D7_0628300	E	E	E
CEPTC_16_0_18_1	PF3D7_0628300	E	E	E
CEPTC_16_0_18_2	PF3D7_0628300	E	E	E
CEPTC_18_1_18_1	PF3D7_0628300	E	E	E
CEPTC_18_1_18_2	PF3D7_0628300	E	E	E
CEPTC_18_2_18_2	PF3D7_0628300	E	E	E
CEPTE_16_0_16_0	PF3D7_0628300	E	E	E
CEPTE_16_0_18_1	PF3D7_0628300	E	E	E
CEPTE_16_0_18_2	PF3D7_0628300	E	E	E
CEPTE_18_1_18_1	PF3D7_0628300	E	E	NE
CEPTE_18_1_18_2	PF3D7_0628300	E	E	E
CEPTE_18_2_18_2	PF3D7_0628300	E	E	E
CHLPCTD	PF3D7_1316600	NE	NE	E
CHOLK	PF3D7_1401800	E	E	NE
CHORS	PF3D7_0623000	E	E	E
CITFUMtm	PF3D7_1223800	NE	NE	NE

CPPPGO	PF3D7_1142400	E	E	E
CSmr	PF3D7_1022500 or PF3D7_0609200	E	E	NE
CTPS2	PF3D7_1410200	E	E	E
CYOOm2	PF3D7_1430900 and PF3D7_0927800 and PF3D7_0928000 and PF3D7_1361700 and mal_mito_2 and mal_mito_1 and PF3D7_1475300 and PF3D7_1010300	E	E	NE
CYOR-mqn4m	PF3D7_0523100 and PF3D7_1426900 and PF3D7_0933600 and PF3D7_1012300 and PF3D7_1462700 and PF3D7_1439400	E	E	NE
CYStec	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	E	E	NE
CYTK1	PF3D7_0111500	E	E	E
DGATpf_16_0_18_0_stcoa	PF3D7_0322300	E	E	E
DGATpf_18_1_18_2_lnlccoa	PF3D7_0322300	E	E	E
DHFR	PF3D7_0417200	E	E	E
DHFS	PF3D7_1324800	E	E	E
DHORD10m	PF3D7_0603300	E	E	NE
DHORtm	PF3D7_1432100	E	E	E
DHPTt	PF3D7_1116500 or PF3D7_0828600	NE	NE	NE
DMATT	PF3D7_1128400	E	E	E
DMPPSyc	PF3D7_0104400	NE	NE	NE
DMQMTm	PF3D7_0724300 or PF3D7_0916600	E	E	E
DOL12PMT	PF3D7_1141600	E	E	NE
DTMPK	PF3D7_1251300	E	E	E
DXPRIic	PF3D7_1467300	E	E	E
DXPSc	PF3D7_1337200	E	E	E
EAR100xc	PF3D7_0615100	E	E	NE
EAR120xc	PF3D7_0615100	E	E	NE
EAR140xc	PF3D7_0615100	E	E	NE
EAR160xc	PF3D7_0615100	E	E	NE
EAR40xc	PF3D7_0615100	E	E	NE

EAR60xc	PF3D7_0615100	E	E	NE
EAR80xc	PF3D7_0615100	E	E	NE
ENO	PF3D7_1015900	E	E	E
ETHAK	PF3D7_1124600	E	E	NE
FACOAL160i	PF3D7_1372400 or PF3D7_0731600 or PF3D7_1477900 or PF3D7_1479000 or PF3D7_0215000 or PF3D7_0215300 or PF3D7_0301000 or PF3D7_0401900 or PF3D7_0525100 or PF3D7_0619500 or PF3D7_1238800 or PF3D7_1253400 or PF3D7_1200700 or PF3D7_0605900	E	E	NE
FACOAL180i	PF3D7_1372400 or PF3D7_0731600 or PF3D7_1477900 or PF3D7_1479000 or PF3D7_0215000 or PF3D7_0215300 or PF3D7_0301000 or PF3D7_0401900 or PF3D7_0525100 or PF3D7_0619500 or PF3D7_1238800 or PF3D7_1253400 or PF3D7_1200700 or PF3D7_0605900	NE	NE	NE
FACOAL181i	PF3D7_1372400 or PF3D7_0731600 or PF3D7_1477900 or PF3D7_1479000 or PF3D7_0215000 or PF3D7_0215300 or PF3D7_0301000 or PF3D7_0401900 or PF3D7_0525100 or PF3D7_0619500 or PF3D7_1238800 or PF3D7_1253400 or PF3D7_1200700 or PF3D7_0605900	NE	NE	NE

FACOAL1821i	PF3D7_1372400 or PF3D7_0731600 or PF3D7_1477900 or PF3D7_1479000 or PF3D7_0215000 or PF3D7_0215300 or PF3D7_0301000 or PF3D7_0401900 or PF3D7_0525100 or PF3D7_0619500 or PF3D7_1238800 or PF3D7_1253400 or PF3D7_1200700 or PF3D7_0605900	NE	NE	E
FACOAL204i	PF3D7_1372400 or PF3D7_0731600 or PF3D7_1477900 or PF3D7_1479000 or PF3D7_0215000 or PF3D7_0215300 or PF3D7_0301000 or PF3D7_0401900 or PF3D7_0525100 or PF3D7_0619500 or PF3D7_1238800 or PF3D7_1253400 or PF3D7_1200700 or PF3D7_0605900	NE	NE	E
FBA	PF3D7_1444800	E	E	E
FCLTm	PF3D7_1364900	NE	NE	NE
FE2t1	PF3D7_0609100	E	NE	E
FMNAT	PF3D7_1015000	NE	NE	E
FPGS	PF3D7_1324800	E	E	E
FRTT	PF3D7_1147500	E	E	E
FRUt1	PF3D7_0204700	E	E	NE
G3PD1	PF3D7_1216200	E	E	NE
G3PD2m	PF3D7_0306400	NE	NE	NE
G3Pthr	PF3D7_0530200 and PF3D7_0508300	E	E	E
GAPD	PF3D7_1462800	E	E	E
GFUCS	PF3D7_1014000	E	E	E
GGTT	PF3D7_0202700	E	E	E
GHMT2r	PF3D7_1235600 or PF3D7_1456100	E	E	E
GK1	PF3D7_0928900 or PF3D7_1251300	E	E	E
GLCt1	PF3D7_0204700	E	E	NE

GLUt5r	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	NE	NE	NE
GLYK	PF3D7_1351600	NE	NE	NE
GLYt	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	NE	NE	NE
GMAND	PF3D7_0813800	E	E	E
GPAM_pf_16_0	PF3D7_1212500 or PF3D7_1318200	E	E	E
GPAM_pf_18_0	PF3D7_1212500 or PF3D7_1318200	E	E	E
GPAM_pf_18_1	PF3D7_1212500 or PF3D7_1318200	E	E	E
GPAM_pf_18_2	PF3D7_1212500 or PF3D7_1318200	E	E	E
GPIAT_18_0_18_1_16_0	PF3D7_0615300	E	E	E
GPIMT12er	PF3D7_1341600	E	E	NE
GRTT	PF3D7_1128400	E	E	E
GSNt	PF3D7_1347200	E	E	NE
GTHOrc	PF3D7_1419800 or PF3D7_0923800	NE	NE	NE
GUAPRTr	PF3D7_1012400	E	E	NE
H2Ot	PF3D7_1132800	E	E	NE
HBZOPTm	PF3D7_0607500	E	E	E
HCO3E	PF3D7_1140000	E	E	E
HEPTT	PF3D7_0202700	E	E	E
HEX1	PF3D7_0624000	E	E	NE
HEX4	PF3D7_0624000	E	E	NE
HEX7	PF3D7_0624000	E	E	NE
HEXTT	PF3D7_0202700	E	E	E
HISt5r	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	E	E	NE
HMBSc	PF3D7_1209600	E	E	NE
HMPK1	PF3D7_0520500	NE	NE	NE

HXPRTr	PF3D7_1012400	E	E	NE
ICDHyrm	PF3D7_1345700	E	E	NE
ILEtec	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	E	E	NE
INSt	PF3D7_1347200	E	E	NE
IPDPSyc	PF3D7_0104400	NE	NE	NE
KAS14c	PF3D7_0626300 or PF3D7_0211400	NE	NE	NE
LEUtec	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	E	E	NE
LIPAMPLm	PF3D7_0923600 or PF3D7_1314600	E	E	E
LIPATPTm	PF3D7_0923600 or PF3D7_1314600	E	E	E
LYSt5r	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	E	E	NE
MAN1PT	PF3D7_1420900	E	E	E
MAN6PI	PF3D7_0801800	NE	E	NE
MANt1	PF3D7_0204700	E	E	NE
MCOATAc	PF3D7_1312000	E	E	NE
MDH7m	PF3D7_0616800	NE	NE	NE
MECDPDH2yc	PF3D7_1022800	E	E	E
MECDPSc	PF3D7_0209300	E	E	E
MEPCTc	PF3D7_0106900	E	E	E
METAT	PF3D7_0922200	E	E	E
METtec	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	E	E	NE
MI3PP	PF3D7_0802500 or PF3D7_0705500	E	E	E
MI3PS	PF3D7_0511800	NE	NE	E

NADH2-mq4m	PF3D7_0915000	NE	NE	NE
NADK	PF3D7_0913300	E	E	E
NADS2	PF3D7_0926700	NE	NE	E
NAMNPP	PF3D7_0629100	NE	NE	E
NDPK1	PF3D7_0605600 or PF3D7_1366500	E	E	NE
NDPK2	PF3D7_1366500 or PF3D7_0605600	E	E	E
NDPK3	PF3D7_1366500 or PF3D7_0605600	E	E	E
NDPK4	PF3D7_1366500 or PF3D7_0605600	E	E	E
NDPK5	PF3D7_1366500 or PF3D7_0605600	E	E	E
NDPK7	PF3D7_1366500 or PF3D7_0605600	E	E	E
NDPK8	PF3D7_1366500 or PF3D7_0605600	E	E	E
NH4t	PF3D7_1132800	E	E	NE
NNAT	PF3D7_1327600	NE	NE	E
OHPHMm	PF3D7_0724300 or PF3D7_0916600	E	E	E
OMBZLMm	PF3D7_0204900 or PF3D7_0407100 or PF3D7_1455200	E	E	E
OMPDC	PF3D7_1023200 or PF3D7_0512700	E	E	E
OMPHHXxm	PF3D7_0815300	E	E	NE
ORNTAr	PF3D7_0608800	NE	NE	NE
P5CRyr	PF3D7_1357900	E	E	NE
PEPthr	PF3D7_0530200 and PF3D7_0508300	E	E	E
PETHCT	PF3D7_1347700	E	E	E
PFK	PF3D7_1128300 or PF3D7_0915400	E	E	E
PGI	PF3D7_1436000	E	E	NE
PHEtec	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	E	E	NE

PIACGT_18_0_18_1	PF3D7_0618900 or PF3D7_1032400 or PF3D7_0935300 or PF3D7_1141400	E	E	E
PIt2m_2	PF3D7_1202200	E	E	NE
PMPK	PF3D7_0520500	NE	NE	NE
PPAP_16_0_16_0	PF3D7_0303200 or PF3D7_0805600	E	E	E
PPAP_16_0_18_0	PF3D7_0303200 or PF3D7_0805600	E	E	E
PPAP_16_0_18_1	PF3D7_0303200 or PF3D7_0805600	E	E	NE
PPAP_16_0_18_2	PF3D7_0303200 or PF3D7_0805600	E	E	E
PPAP_18_0_18_0	PF3D7_0303200 or PF3D7_0805600	E	E	E
PPAP_18_0_18_1	PF3D7_0303200 or PF3D7_0805600	E	E	NE
PPAP_18_1_18_2	PF3D7_0303200 or PF3D7_0805600	E	E	E
PPAP_18_2_18_2	PF3D7_0303200 or PF3D7_0805600	E	E	E
PPAc	PF3D7_0316300 or PF3D7_1235200 or PF3D7_1456800	E	E	E
PPAm	PF3D7_0316300 or PF3D7_1235200 or PF3D7_1456800	E	E	E
PPBNGSc	PF3D7_1440300	E	E	NE
PPC	PF3D7_1426700	NE	NE	NE
PPM	PF3D7_1012500	NE	NE	NE
PPPGO6m	PF3D7_1028100	NE	E	NE
PPTT	PF3D7_0202700	E	E	E
PRPPS	PF3D7_1327800 or PF3D7_1325100	E	E	E
PSCVT	PF3D7_0206300	NE	NE	E
PSD_18_1_18_1	PF3D7_0927900	E	E	NE
PSSA_18_0_18_1	PF3D7_1366800	E	E	E
AGPAT1_18_0_20_4	PF3D7_1444300	E	E	E
CDS_18_0_20_4	PF3D7_1409900	E	E	E
PSSA_18_0_20_4	PF3D7_1366800	E	E	E
PSSA_18_1_18_1	PF3D7_1366800	E	E	E
PSSA_18_1_20_4	PF3D7_1366800	E	E	E
PSSA_20_4_20_4	PF3D7_1366800	E	E	E

PUNP3	PF3D7_0513300	NE	NE	NE
PUNP5	PF3D7_0513300	NE	NE	NE
PYDXK	PF3D7_0616000	E	E	NE
PYK	PF3D7_0626800	E	E	NE
PYKc	PF3D7_1037100	E	E	E
PYNP2r	PF3D7_0513300	NE	E	NE
PYRt2m	PF3D7_1470400 or PF3D7_1340800	E	E	NE
RBFK	PF3D7_1359100	NE	NE	E
RNDR1	PF3D7_1437200 and PF3D7_1405600 and PF3D7_1015800 and PF3D7_1457200	E	E	NE
RNDR2	PF3D7_1437200 and PF3D7_1405600 and PF3D7_1015800 and PF3D7_1457200	E	E	NE
RNDR3	PF3D7_1437200 and PF3D7_1405600 and PF3D7_1015800 and PF3D7_1457200	E	E	NE
RNDR4	PF3D7_1437200 and PF3D7_1405600 and PF3D7_1015800 and PF3D7_1457200	E	E	NE
RPE	PF3D7_1219900	E	E	NE
RPI	PF3D7_0514600	E	E	NE
SERPT	PF3D7_1415700	E	E	E
SHKK	PF3D7_0206300	NE	NE	E
SUCOASm	(PF3D7_1437700 or PF3D7_1431600) and PF3D7_1108500	NE	NE	NE
THMP	PF3D7_0715000	NE	NE	NE
THRt5r	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	E	E	NE
TKT1	PF3D7_0610800	E	E	NE
TKT2	PF3D7_0610800	E	E	NE
TMDS	PF3D7_0417200	E	E	E
TMPKr	PF3D7_1251300	E	E	NE
TPI	PF3D7_1439900 or PF3D7_0318800	E	E	NE

TRPt	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	E	E	NE
TYRt	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	E	E	NE
UAGDP	PF3D7_1343600 or PF3D7_0517500	E	E	E
UMPK	PF3D7_0111500	E	E	E
UPP3Sc	PF3D7_1209600	E	E	NE
UPPDC1c	PF3D7_0607300	NE	NE	NE
URIt	PF3D7_1347200	E	E	NE
VALtec	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	E	E	NE
CBPSam	PF3D7_1308200	E	E	NE
PIt2r	PF3D7_1340900	E	E	NE
FRD_mt	PF3D7_1034400 and PF3D7_1212800 and PF3D7_0611100	E	E	NE
GTHOXti	PF3D7_0112200 or PF3D7_1229100	NE	NE	NE
SERGLYexR	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	E	E	NE
DSAT_pmtcoa	PF3D7_0508200 or PF3D7_1403700	NE	NE	E
DSAT_stcoa	PF3D7_0508200 or PF3D7_1403700	NE	NE	E
DSAT_lignocoa	PF3D7_0508200 or PF3D7_1403700	NE	NE	E

FACOAL240i	PF3D7_1372400 or PF3D7_0731600 or PF3D7_1477900 or PF3D7_1479000 or PF3D7_0215000 or PF3D7_0215300 or PF3D7_0301000 or PF3D7_0401900 or PF3D7_0525100 or PF3D7_0619500 or PF3D7_1238800 or PF3D7_1253400 or PF3D7_1200700 or PF3D7_0605900	NE	NE	E
SMSg_18_1_18_0_pchol_18_1_18_1	PF3D7_0625100 or PF3D7_0625000	E	E	NE
SMSg_18_1_24_0_pchol_18_2_18_2	PF3D7_0625100 or PF3D7_0625000	E	E	NE
CDS_18_1_20_4	PF3D7_1409900	E	E	E
AGPAT1_18_1_20_4	PF3D7_1444300	E	E	E
CDS_20_4_20_4	PF3D7_1409900	E	E	E
AGPAT1_20_4_20_4	PF3D7_1444300	E	E	E
PSD_20_4_20_4	PF3D7_0927900	E	E	NE
AGPAT1_16_0_20_4	PF3D7_1444300	E	E	E
PPAP_16_0_20_4	PF3D7_0303200 or PF3D7_0805600	E	E	E
CEPTC_16_0_20_4	PF3D7_0628300	E	E	E
GPAM_pf_20_4	PF3D7_1212500 or PF3D7_1318200	E	E	E
CEPTC_16_0_18_0	PF3D7_0628300	E	E	E
PGPPT_pf	PF3D7_0820200	E	E	E
CLPNSm_pf	PF3D7_0609400	E	E	E
THZPSN	(PF3D7_0716600 or PF3D7_0727200) and (PF3D7_1365400 or PF3D7_1333200)	E	E	NE
FAS180COA	PF3D7_0920000 or PF3D7_0605900 or PF3D7_0109300	E	E	NE
TMPPP	PF3D7_0614000	NE	E	NE
PDH_like_mediated_by_BCKDH	PF3D7_0504600 and PF3D7_1232200 and PF3D7_0303700 and PF3D7_1312600	E	E	NE
DGATpf_18_0_18_1_pmtcoa	PF3D7_0322300	E	E	E
DGATpf_18_0_18_0_ocdcea	PF3D7_0322300	E	E	E

SMSg_18_1_16_0_pchol_18_1_18_2	PF3D7_0625100 or PF3D7_0625000	E	E	NE
THBPT4ACAMDASE	PF3D7_1108300	NE	E	NE
ACS	PF3D7_0627800	E	E	NE
AKGOAAtm	PF3D7_0823900	E	E	NE
ASPTA	PF3D7_0204500	NE	NE	NE
CITtcm	PF3D7_1223800	NE	E	NE
DHORTS	PF3D7_1472900	NE	E	E
GLNt5r	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	NE	NE	NE
GLUDyc	PF3D7_1416500 or PF3D7_1430700	NE	NE	NE
GLYCt	PF3D7_1132800	E	E	NE
GTHOrh	PF3D7_1419800	E	E	NE
GTHP_api	PF3D7_1212000 or PF3D7_0729200	E	E	NE
GUAt	PF3D7_1347200	E	E	NE
HYXNt	PF3D7_1347200	E	E	NE
L-LACt2r	PF3D7_0926400 or PF3D7_0210300	E	E	NE
LDH_L	PF3D7_1325200 or PF3D7_1324900	E	E	NE
MDH	PF3D7_0618500	E	E	NE
NNAMr	PF3D7_0320500	E	E	NE
ORPT	PF3D7_0512700 or PF3D7_1023200	E	E	E
PGK	PF3D7_0922500	E	E	E
PGM	PF3D7_0413500 or PF3D7_1120100	E	E	E
PLPS1	PF3D7_1116200 and PF3D7_0621200	E	E	NE
PMANM	PF3D7_1017400	E	E	E
PROt5r	PF3D7_0629500 or PF3D7_1208400 or PF3D7_1231400 or PF3D7_0209600 or PF3D7_0515500 or PF3D7_1132500	NE	NE	NE
SUCOAS1m	(PF3D7_1437700 or PF3D7_1431600) and PF3D7_1108500	NE	E	NE

THD1m	PF3D7_1453500	E	E	NE
THIORDX	PF3D7_1212000 and ((PF3D7_0802200 and PF3D7_1457200) or PF3D7_1457200 or PF3D7_1438900)	E	E	NE
URAt	PF3D7_1347200	E	E	NE
URIDK2r	PF3D7_1251300	E	E	NE
orottm	PF3D7_1432100	E	E	E
PYDXDH	PF3D7_1364600 or PF3D7_1409100	E	E	NE
PPKr	PF3D7_1230200	NE	NE	NE