In silico Identification of Toxins and Their Effect on Host Pathways: Feature Extraction, Classification and Pathway Prediction

A thesis submitted to Indian Statistical Institute in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

By

Rishika Sen

Supervisor: Professor Rajat K. De

Machine Intelligence Unit
Indian Statistical Institute
Kolkata - 700 108, India
January 2, 2021

Ma and Baba

Acknowledgement

A Ph.D. thesis is not just the collection of original research works carried out by a person for obtaining a Ph.D. degree, but an amalgamation of the assistance, guidance, and constant encouragement provided by several persons. I would, therefore, like to convey my sincere thanks to all my teachers, family, and friends, without whom the thesis would never see the light of the day.

First, I would like to express my heartfelt gratitude and indebtedness to my supervisor Professor Rajat Kumar De. It is not possible to describe in a few words, the immeasurable supports and invaluable suggestions he has provided in my research works. It was he, who first introduced me to the world of bioinformatics and computational biology, which, in turn, encouraged me to pursue my Ph.D. degree in Computer Science. Along with research problems, he has also guided me to handle difficult scenarios in my personal life.

I would like to convey my sincere gratitude to the Machine Intelligence Unit, Indian Statistical Institute, Kolkata for providing me with a great research environment to pursue my Ph.D. degree. I want to thank all my teachers from M.Sc. and B.Sc. degrees at University of Calcutta; without their teachings and blessings, I would not be able to complete this thesis.

I am indebted to the Dean of Studies and the Director of the Indian Statistical Institute (ISI) for providing me the fellowship, travel grants, and after all a good academic environment. I express my sincere thanks to the authorities of ISI for the facilities extended to carry out the research work and for providing me every support during my tenure. I would also like to acknowledge all the timely supports that I have received from the office staff of our institute during the tenure of my Ph.D.

I am extremely thankful to my co-authors, Dr. Somnath Tagore and Dr. Losiana Nayak for providing help when needed. Their valuable suggestions have helped for the betterment of my work. I am thankful to my seniors and colleagues for their unconditional support.

My biggest thanks to Alexandra Elbakyan, the creator of Sci-Hub. This journey would not have been possible without her. I am greatly indebted to my parents and my family. I specially thank my aunt and my grandmother for their support. I would like to express my gratitude to all my colleagues and alumni in Machine Intelligence Unit for their constant encouragement, support, and friendship. I wholeheartedly thank my friends Arindam Pal, Diptavo Dutta, Indrani Ray, Kushal Sen, Mohar Mukherjee, Monalisa Pal, Poulami Pal, Sampa Misra, and Tanmay Mitra for their unconditional love and support. I thank my seniors Dr. Abhijit Dasgupta and Dr. Kaustuv Nag for helping me out when needed. Last but not the least, I want to thank everyone whom I might have missed here, for their good wishes and support.

Indian Statistical Institute — Rishika Sen

Publications

This dissertation is a culmination of my research work at the Machine Intelligence Unit at the Indian Statistical Institute, Kolkata during the period 2014–2020. I hope that all the experience that I have gained during this period is adequately reflected in the thesis. Following are the list of publications that have been used in the thesis. Chapter 2 is based on the article [354]. Chapter 3 is constructed from [353]. Chapter 4 is inspired by [355]. Chapter 5 is based on [351]. Chapter 6 is inspired by [356]. Chapter 7 is constructed from [352].

Article

- **Rishika Sen**, Somnath Tagore, Rajat Kumar De. "Cluster Quality based Non-Reductional (CQNR) oversampling technique and Effector Protein Predictor based on 3D structure (EPP3D) of proteins." *Computers in Biology and Medicine*, vol. 112, no. 103374, pp. 1–13, 2019.
  doi: 10.1016/j.compbiomed.2019.103374, SCI indexed. [355]
- **Rishika Sen**, Losiana Nayak, and Rajat Kumar De. "PyPredT6: A Python-based Prediction Tool for Identification of Type VI Effector Proteins." *Journal of Bioinformatics and Computational Biology*, vol. 17, no. 03, pp. 1–18, 2019.
  doi: 10.1142/S0219720019500197, SCI Indexed. [353]
- **Rishika Sen**, Somnath Tagore, Rajat Kumar De. "ASAPP: Architectural Similarity-based Automated Pathway Prediction System and its Application in Host-Pathogen Interactions." *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 17, no. 2, pp. 506–515, 2020.
  doi: 10.1109/TCBB.2018.2872527, SCI Indexed. [356]
- **Rishika Sen**, Losiana Nayak, and Rajat Kumar De. "A review on host-pathogen interactions: classification and prediction." *European Journal of Clinical Microbiology & Infectious Diseases*, vol. 35, no. 10, pp. 1581–1599, 2016.
  doi: 10.1007/s10096-016-2716-7, SCI Indexed. [354]
- **Rishika Sen**, Rajat Kumar De. "DeepT7: A Deep Neural Network System for Identification of Type VII Effector Proteins.", Computational Biology and Chemistry (under

revision). [351]

- **Rishika Sen**, Rajat Kumar De. “Boolean logic-based Network Robustness Analyzer (BNRA) and its application to a system of Host-Pathogen interactions.”, (under preparation). [352]

Poster

- **Rishika Sen**, Losiana Nayak, and Rajat Kumar De. ”Classification, Prediction and Analysis of Type VI Secreted Effector Proteins”. Advanced Lecture Course - Molecular Mechanisms of Host-Pathogens Interactions and Virulence in Human Fungal Pathogens, University of Aberdeen, Nice, France. (2017).
- **Rishika Sen**, Losiana Nayak, and Rajat Kumar De. ”Signature Pattern Mining of Type VI effector Proteins”. EMBO Global Exchange Lecture Course: Malaria Genomics and Public Health. (2017), doi: 10.13140/RG.2.2.15231.61601.

Workshops attended

- India|EMBO Symposium : Regulatory epigenomics: From large data to useful models, March 10 to 13, 2019, Chennai, India.
- Advanced Lecture Course - Molecular Mechanisms of Host-Pathogens Interactions and Virulence in Human Fungal Pathogens, May 13 to 19, 2017, University of Aberdeen, Nice, France.
- International Symposium on Health Analytics and Disease Modeling (HADM 2016), 29th February & 1st March, 2016, Indian Institute of Public Health, Hyderabad (IIPHH), India.
- 3rd Institute of Mathematical Sciences Workshop and Conference on Modeling Infectious Diseases, November 23 to December 1, 2015, Chennai, India.

Online repository

The algorithms developed in this thesis have been transformed into standalone systems coded in Python/MATLAB for the convenience of further research. The links are as follows:

List of repositories

Tool	Language	Website
PyPredT6	Python	http://projectphd.droppages.com/PyPredT6.html
EPP3D	Python	http://projectphd.droppages.com/CQNR.html
CQNR	Python	http://projectphd.droppages.com/CQNR.html
DeepT7	Python	http://projectphd.droppages.com/DeepT7.html
ASAPP	MATLAB	http://asapp.droppages.com/
BNRA	MATLAB	http://projectphd.droppages.com/BNRA.html

Abstract

Identification of toxins, which are either proteins or small molecules, from pathogens is of paramount importance due to their crucial role as first-line invaders infiltrating a host, often leading to infection of the host. These toxins can affect specific proteins, like enzymes that catalyze metabolic pathways, affect metabolites that form the basis of metabolic reactions, and prevent the progression of those pathways, or more generally they may affect the regular functioning of other proteins in signaling pathways in the host. In this regard, the thesis addresses the problem of identification of toxins, and the effect of perturbations by toxins on the host pathways based on three tasks: feature extraction, classification and pathway prediction. The thesis starts with *in silico* identification of such toxins in pathogens. This is followed by the analysis of the effect of toxins on various metabolic and signaling pathways of the host.

Identification of effector proteins has been achieved using feature extraction and classification techniques. A lot of work has been done in the prediction of Type III and Type IV effector proteins based on their primary structure. However, this is not the case for Type VI effector proteins. In this regard, the thesis first introduces a novel framework for fast and accurate identification of Type VI effector proteins based on their primary and secondary structures. While working on Type VI effectors, it came into our attention that no attempts have been made for prediction of effectors based on their three-dimensional structure. This thesis introduces a unique set of three-dimensional structural features and builds a novel predictor using them. Since the effector protein dataset was unbalanced, we have introduced a novel algorithm for oversampling of an unbalanced biological dataset, which does not eliminate samples as noise and ensure generation of synthetic samples strictly in the vicinity of the minority class samples. Integrating the unique feature set and the oversampling algorithm, a novel effector protein predictor has been developed. Due to the unavailability of three-dimensional structure of Type VII effector proteins and their importance in spreading pathogenesis in hosts, we have developed a deep neural network-based system to uniquely identify Type VII effectors. The system identifies effectors based on the primary and secondary structure of Type VII effectors.

Identification of toxins remains incomplete if their effect on host is not investigated. In this regard, along with identification of toxins, analysis of the effect of perturbations on various pathways by the novel algorithms has been furnished in the thesis. A new structure-based automated metabolic pathway prediction algorithm has been introduced, which predicts a probable pathway considering a set of metabolites. This algorithm has been applied to metabolic pathways of the hosts to study the effect of toxins on them. Apart from metabolic pathways, toxins also affect signaling pathways. This perturbation has been studied, and a novel algorithm has been developed to quantify the effect of the perturbation on these signal-

ing pathways. Overall, this thesis is dedicated to the design of computational algorithms to identify the toxins secreted by pathogens and the effect of these toxins on the host pathways.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction and Scope of the Thesis

1.1 Introduction

Infectious diseases played an undeniably significant role in human history. The continual expansion of human population has led to recurrent invasion by increasing number of various pathogens in human population. The appearance of new pathogens has led to the occurrence of new diseases, some of which have been proved to be lethal [256]. A current example in this regard is COVID-19 due to sudden emergence of a novel pathogen, called SARS-CoV-2. More than 30 million people across more than 200 countries have been infected with SARS-CoV-2, out of which about 1 million people have died. The condition is severe for USA, Brazil and India. The number of infected persons in India is around 6 million, while we have lost about 1 lakh citizens, till September 2020.

With the advancement in the field of biological and medical sciences, and in healthcare, accompanied by the invention of new experimental devices and methods [255], new pathogens, their biological characteristics and their effect on various hosts are being discovered and analyzed. The need for a precise understanding of the lifecycle of such pathogens, their invasion techniques, and finally, the outcome of their invasion in the body of the host is crucial [160]. While it has been possible to determine the cure for many diseases, like polio [370], diptheria [21], tetanus [31], through years of extensive research, the cure for some, like AIDS, dengue, common cold and herpes simplex still remains unknown.

The control and prevention of infectious diseases are likely to be increasingly dependent on a solid understanding of the molecular mechanism of pathogens [3]. The effort to understand pathogens is being carried out for decades. Understanding the molecular skeleton of pathogens includes exploring their genomes, proteomes and different variants [399], and thereby, unraveling the 3D structure of proteins. This is crucial since the structure of a protein has a significant impact on its function.

With such an enormous array of pathogens infecting humans and other animals, computational methods have found new ways to facilitate the study of pathogens and infection. As new pathogens are being discovered every day, the demand to find a cure in a short amount of time is also elevated. For example, consider the recently discovered disease COVID-19. The disease spread over more than 200 countries within a year, killing nearly 1 million people. This virus has been proved to be lethal to the older people [302] and people with comorbidity [437]. Discovery of drugs and vaccines for this disease is an immediate necessity.

The study of infection caused by pathogens encompasses many diverse aspects of modern science, including computational biology, bioinformatics, and systems biology. Bioinformatics assisted biosurveillance and early warning have been designed to predict infectious disease outbreaks [367]. Such a framework has been developed by combining the genetic and geographic data of pathogens to facilitate determining its origin, and recognizing the migration routes through which the strains spread regionally and globally. The biosurveillance and microbial profiling focused text mining tools assist in infectious disease outbreak detection [367]. It is based upon bioinformatics models, which include the timeliness of outbreak detection and accuracy. Another utility that bioinformatics finds in disease-based research is the prediction of protein-protein interactions between pathogens and their hosts, which facilitates understanding of the infection mechanism [362]. Computational protein structure prediction methods provide crucial information on a large number of sequences whose structures have not yet been determined experimentally [26]. From molecular level to population level, bioinformatics and computational biology have facilitated the research of diseases and consequently have further helped in designing synthetic drugs [43].

In this thesis, we have developed *in silico* methods to identify pathogenic bacterial toxins that disrupt the normal cellular functionality in a host. Feature extraction and classification techniques have been incorporated to develop systems that are capable of accurately identifying such toxins. Additionally, we have designed algorithms based on structural characteristics of metabolites, to predict unknown pathways, and how these pathways are perturbed in the presence of such toxins. Algorithms have also been developed to quantify the stability of pathways and to demonstrate how the stability is affected by perturbation through toxins.

Pathogens infect hosts primarily in four stages: invasion, evasion, replication, and elimination [354]. In this thesis, we primarily concentrate on the first stage of attack of bacterial pathogens on their hosts, i.e., invasion. In this stage, bacteria invade the hosts and liberates toxins into them. Here, we have developed new *in silico* algorithms and methods by extracting novel features, which facilitate the identification of such toxins. Toxins being released into the host systems disrupt the biochemical pathways of the hosts. In regard to this, we have developed *in silico* methods to understand the effect of such toxins on the pathways of the hosts.

1.2 Basic concepts

At the molecular and cellular levels, pathogens can infect the hosts by secreting toxins, which cause symptoms to appear. Infecting the hosts leads to the disruption of homeostasis in their systems. For detection and prevention of such occurrences, a thorough understanding of how a pathogen invades a host is crucial. Achieving such a goal is feasible in real-time conveniently by building computational algorithms and methods. However, in order to build computational algorithms that would mimic the effect of a pathogen on a host, one should have an in-depth understanding of the underlying biological mechanisms. In this section, we take a look at the basic concepts of molecular biology, and get acquainted with various terminologies, like pathogen, pathogenicity, and host-pathogen interactions, among others.

1.2.1 Some terms in molecular biology

Molecular biology deals with the molecular basis of biological activity being carried out in an organism. This study includes the interactions among DNA, RNA, proteins, their biosynthesis, and the regulation of these interactions. DNA effectively encode genetic information which is made available to the organism in the form of proteins. The process by which information encoded in DNA is conveyed/propagated into proteins is called the central dogma.

Central dogma The central dogma of molecular biology states how the instructions encoded in DNA are propagated to a newly formed functional product. It is described as the flow of genetic information from DNA to RNA (through transcription), and finally to make a functional product, i.e., a protein (through translation).

DNA and RNA Deoxyribonucleic acid (DNA) is a carrier of genetic information for development, function, growth, and reproduction of all organisms. It is a molecule composed of two chains that coil around each other to form a double helix. Ribonucleic acid (RNA) is a polymeric molecule whose primary role is to carry information from DNA for protein synthesis. RNA is in the form of a chain of nucleotides. However, unlike DNA, it is more often found in nature as a single-strand. DNA is made up of four nucleobases, viz., adenine (A), cytosine (C), guanine (G), and thymine (T), while RNA is composed of adenine (A), cytosine (C), guanine (G), and uracil (U). These nucleobases are called primary units. They function as the fundamental building blocks of genes.

Gene A gene is the basic physical and functional unit of heredity. A sequence of nucleotides in DNA, which codes for a protein molecule is termed as a gene. However, not all genes code for proteins. In humans, genes vary in size from a few hundred DNA bases to more than 2 million bases.

Protein Proteins are large macromolecules consisting of long chains of amino acid residues. They perform a diverse set of functions within organisms, which includes DNA replication, catalyzing metabolic reactions, providing structure to cells and organisms, responding to stimuli, and transporting molecules from one location to another, among others. Different proteins have different amino acid sequences.

Amino acids are basic units of a protein. There are 20 different amino acids, some of which combine into peptide chains (polypeptides) to form the building blocks of a vast array of proteins. These twenty amino acids include Alanine (A), Arginine (R), Asparagine (N), Aspartic acid (D), Cysteine (C), Glutamine (Q), Glutamic acid (E), Glycine (G), Histidine (H), Isoleucine (I), Leucine (L), Lysine (K), Phenylalanine (F), Methionine (M), Serine (S), Proline (P), Threonine (T), Tyrosine (Y), Tryptophan (W) and Valine (V). There are four well-defined levels of protein structure, as stated below.

Primary structure: The primary structure of a protein being linear is represented by the sequence of amino acids in the polypeptide chain. It is represented in the form of a series of amino acids like "... MKLPHSTYV ...".

Secondary structure: Secondary structure refers to highly regular local sub-structures on the actual polypeptide backbone chain. It is an intermediate stage before a protein gets folded into a three-dimensional tertiary structure.

Tertiary structure: Tertiary structure refers to the three-dimensional structure of protein molecules. It is represented by the coordinates of each of the atoms forming the protein molecule. It is also known as the three-dimensional structure.

Quaternary structure: Many proteins are made up of a single polypeptide chain and thus have only three levels of structure, as discussed above. However, some proteins are made up of multiple polypeptide chains, also known as subunits. When these subunits come together, they give the protein its quaternary structure. Quaternary structure is the three-dimensional structure consisting of the aggregation of two or more individual polypeptide chains (subunits) that operate as a single functional unit.

Protein structures play a crucial role in the functionality of protein molecules [304]. Three-dimensional structure of a protein defines its size, shape, and function. For example, one characteristic that affects function is the hydrophobicity of a protein, which is determined by the primary and secondary structures [131]. Cell membranes contain large amount of extremely hydrophobic lipids. The membrane-spanning regions of membrane proteins are typically alpha-helices, made of hydrophobic amino acids. These hydrophobic regions interact favorably with the hydrophobic lipids in the membrane, forming stable membrane structures. The folding of a protein facilitates interactions among amino acids that may be distant from each other in its primary sequence of amino acids [225].

One of the most promising developments achieved by the study of human genes and proteins is the identification of potential new drugs for treatment of diseases. This relies on proteome and genome information to identify proteins associated with a disease. With such crucial information, computer software can be used to design possible targets for new drugs. For example, if a particular protein is implicated in a disease, its 3D structure provides information on the type of protein structure it will be able to bind to. A computer algorithm can be developed that designs molecules (drugs) with structures complementary to the disease protein to block its action. A molecule that fits into the active site of an enzyme, but cannot be released by the enzyme, deactivates the enzyme. This concept is the basis of new drug-discovery tools, which aim to find new drugs to deactivate proteins mediating a disease.

1.2.2 Pathogen

A pathogen is an organism that enters into another organism (called host) and can cause disease in the latter organism [10]. Usually, the term ‘pathogen’ is used to describe an infectious microorganism or agent, such as a bacterium, virus, prion, protozoan, fungus, or viroid. Different pathogens have different ways of invading hosts. For example, bacterial pathogens invade hosts via proteins, while viruses invade by RNA.

Diseases caused by infectious agents are known as pathogenic diseases. For example, cholera is caused by bacteria, while HIV and COVID-19 by virus, Creutzfeldt-Jakob disease by prions, malaria by protozoan, Aspergillosis by fungus, and hepatitis D is caused by viroid. However, not all diseases are caused by pathogens. Some diseases are hereditary. An example of such a disease is Huntington’s disease, which is caused by the inheritance of abnormal genes.

Bacteria can be classified into two groups based on the structure of their cell wall. These two groups are gram-positive and gram-negative bacteria. Gram-positive bacteria have a thick peptidoglycan layer and no outer lipid membrane whilst gram-negative bacteria have a thin peptidoglycan layer and have an outer lipid membrane. The difference in the structure of cell wall makes gram-positive bacteria more susceptible to antibiotics, while making gram-negative bacterias more resistant to them.

1.2.3 Pathogenicity

Pathogenicity is the potential disease-causing capacity of pathogens in host systems [198]. A pathogen is described in terms of its ability to enter tissue of a host, produce toxins, hijack nutrients of the host, reproduce, colonize, and immunosuppress the host. Toxins are poisonous substances produced by various bacteria. They can be small molecules or proteins that are capable of causing disease. Toxins can be classified as either exotoxins (being excreted by an organism), or endotoxins (being released mainly when bacteria are lysed).

Secretion system: Bacterial pathogens primarily invade hosts via protein secretion [10]. Pathogens, particularly the gram-negative bacteria, have nanomachines to secrete various virulence factors across the bacterial cell envelope. Such nanomachines are known as secretion systems [93]. Bacterial secretion systems are protein complexes present on the cell membranes of bacteria, which facilitate secretion of toxins into hosts. These secretion systems release proteins (exotoxins), called effectors, into the body of hosts when they come in contact with them [93,102]. The secretion system of gram-negative bacteria can be classified as Type I, Type II, Type III, Type IV, Type V, Type VI, Type VIII [110], and Type IX [254]. Type VII secretion system has been discovered in gram-positive bacteria [2].

Types of interactions: The relationship between a host and a pathogen in the host system is dynamic since one modifies the activities and functions of the other [397]. This relationship is termed as host-pathogen interactions [68], the mechanism by which microbes or viruses sustain themselves within host organisms at molecular, cellular, organismal, or population level [66]. The consequence of such a relationship depends on the relative degree of resistance or susceptibility of the host and the virulence of the pathogen; mainly due to the effectiveness of the host defense mechanisms.

There exist three types of host-pathogen interactions. How a pathogen interacts with a host, decides what sort of interaction it is [67]. An interaction where a pathogen is benefitting from a host while the host is not affected by the interaction is termed as a commensal relationship. An example of this is bacteroides, which resides in the human intestinal tract but provides no known benefit or harm. The interaction by which both a pathogen and a host benefit from, as seen in human stomach, is termed as mutualism. Bacterial phyla, viz., firmicutes, bacteroidetes, actinobacteria, and proteobacteria, assist in breaking down nutrients for host, and in return, the host body acts as their ecosystem. Interactions by which pathogens benefit from their hosts while hosts are harmed, are recognized as parasitism. This can be seen in the unicellular parasite *Plasmodium falciparum*, which causes malaria in human.

Pathogenic variability in hosts: Context-dependent pathogenicity [33] is a term used to describe a characteristic of pathogens where their disease-causing capacity varies by the genetic and environmental factors of the host that a pathogen finds itself in. One example of pathogenic variability in *Homo sapiens* is that involving *Escherichia coli* as the pathogen. Normally, these bacteria flourish as normal and healthy microbiota in the intestine. However, if *E. coli* relocates to a different region of the digestive tract of the body, it can cause intense diarrhea. Some strains of a pathogen are less virulent than some other strains. For example, in *Sclerotinia trifoliorum*, a degenerate non-virulent strain of the pathogen produces more protopectinase (the quantity of protopectinase being a measure of pathogenicity) than a normal strain, but only the normal strain secretes a toxin and is considered virulent. In the *Mycobacterium* genus, *Mycobacterium smegmatis* is a nonpathogenic Mycobacterium, while *Mycobacterium leprae* is a pathogenic species causing the disease leprosy [325].

1.3 Importance of computer science in prediction, identification and prevention of diseases

Researchers are aiming to understand genetic variability and how it contributes to pathogen interactions and variability within a host. They are also trying to limit the transmission methods for many pathogens to prevent rapid spread in hosts. In order to cope with the changing pathogenic environment, treatment methods need to be revised to deal with drug-resistant microbes. With new deadly diseases being discovered every day, along with an array of pathogens, experimental analysis of such diseases and pathogens is time-consuming. Bioinformatics and computational biology come into rescue by reducing the search space, and thereby making the analysis time-efficient to a great extent.

Computational exploration for solving biological problems is what constitutes the fields of Bioinformatics, Computational Biology, and Systems Biology. These fields play a significant role in expanding our knowledge in modern biology with the generation of various datasets dealing with different aspects of biological systems. These datasets include those on sequences of nucleotides/amino acids in genes/proteins, gene expression, protein-protein interactions, and host-pathogen interactions. Alignment methods, machine learning, and structural analysis methods are all essential for understanding different biological processes, including that involving host-pathogen interactions. How such interactions can be exploited to find a cure for an associated disease is the main challenge to understand. The development of vaccines, novel drugs, and other therapeutics are highly dependent on the knowledge gained from investigating host-pathogen interactions. As mentioned above, the involvement of computer science in the field of biology has led to the emergence of three interdisciplinary fields of study, viz. bioinformatics, computational biology, and systems biology.

1.3.1 Bioinformatics

Bioinformatics is an interdisciplinary study that involves development of algorithms and methodology to extract knowledge from biological data [248]. Being an interdisciplinary field, bioinformatics combines computer science, biology - particularly molecular biology, mathematics, information technology and statistics to analyze and interpret biological data to predict and identify diseases, and to design rational drugs. Bioinformatics deals with *in silico* analyses of biological queries using statistical, mathematical and computational techniques [431]. Current studies of bioinformatics include analysis of DNA sequence, gene and protein expression, and cellular organization [62].

The field of bioinformatics has become indispensable in the study of modern biology. Techniques, developed under the umbrella of this field, facilitate extraction of significant amount of knowledge from a large volume of raw data generated through high throughput technology and experimental molecular biology [19]. In genetics, the study helps in annotating and sequencing genomes, and their observed variants. It plays a major role in mining biological literature and the development of gene ontologies to organize and query biological data [28]. It also has a significant impact on the analysis of protein and gene expression and regulation. The field also facilitates comparing, analyzing, and interpreting genomic and genetic data, and more generally, in the understanding of evolutionary aspects of molecular biology [315]. In structural biology, bioinformatics helps in determining structure of DNA [363], RNA [100], proteins [220] as well as biomolecular interactions [425].

1.3.2 Computational Biology

Computational Biology uses a combination of biology and information sciences [417] to develop models that help understand biological processes and relationships from biological data. Experimental data such as sequences, images, and concentrations of biomolecules are used as input to develop models to predict the behavior of biological systems. These models may help in describing the vital tasks carried out by particular nucleic acid or peptide sequences, identifying the genes whose expression produces a particular behavior, determining the changes in gene/protein expression or localization leading to a particular disease, and describing the changes in cell organization influencing cellular behavior.

1.3.3 Systems Biology

Systems biology is the interdisciplinary branch of modeling complex biological systems with the help of computational and mathematical techniques [218]. It involves the study of interactions among the components of complex biological systems, and how these interactions influence the functionality and the behavior of such systems [224]. It seeks to study biological systems as a whole. The Human genome project was an outcome of the application of systems biology. This led to the emergence of collaborative ways of working on problems in genetics. This field of study helps in better understanding of the processes that are going on in biological systems in entirety. Identification of gene regulatory logic in biochemical networks, stochastic modeling of intricate biological systems, and systems biology in drug discovery are some of the challenging avenues of systems biology research.

Bioinformatics came into picture in the early 1970s. It has been identified as the technology of incorporating informatics in understanding various biological systems. With time, computational biology has become an important part of modern biology [306]. Computational biology has been used to sequence the human genome, create accurate models of the human brain, and to assist in modeling biological systems. Systems biology has gained attention, particularly from the year 1999. Specifically, the NIH defines Computational biology, Bioinformatics [194] and Systems Biology [415] as follows.

- **Computational biology:** “The development and application of data-analytical and theoretical methods, mathematical modeling and computational simulation techniques to the study of biological, behavioral, and social systems.”
- **Bioinformatics:** “Research, development, or application of computational tools and approaches for expanding the use of biological, medical, behavioral or health data, including those to acquire, store, organize, archive, analyze, or visualize such data.”
- **Systems Biology:** “An approach in biomedical research in understanding the larger picture - be it at the level of the organism, tissue, or cell - by putting its pieces together. It is in stark contrast to decades of reductionist biology, which involves taking the pieces apart.”

These three areas/concepts together can be described as the research, development and application of *in silico* algorithms and tools for modeling, analysis, and prediction of biological systems.

1.3.4 Importance of bioinformatics, computational biology and systems biology in the study of host-pathogen interactions: feature extraction, classification and pathway prediction

Understanding host-pathogen interactions is a crucial step to unravel mechanisms of infectious disease, as well as its prediction, prevention, and treatment [326]. The analysis of different stages of infection throws light on the mechanisms by which pathogens invade and replicate in their hosts. Pathogens invade hosts by secreting toxins into host bodies. Toxins are mainly proteins synthesized in pathogens and liberated into hosts, which damage host systems. The other proteins in pathogens are house-keeping proteins helping in day to day survival of pathogens [142].

Thus, identification of toxins forms an essential task that aids in rational drug design. These toxins provide three-dimensional templates for creating small molecules that mimic the toxins with interesting pharmacological properties. They can also be used as pharmacological tools to uncover potential therapeutic targets [179]. In other words, in order to develop rational drugs, it is vital to know the structure and function of the proteins disrupting homeostasis of hosts. Drugs would introduce/induce new proteins into hosts, which may bind to the toxins and render them neutral and ineffective [311]. Given such a vast array of pathogens and the variety of toxins secreted by them along with thousands of housekeeping proteins existing in them, it is time-consuming and expensive to check experimentally every protein of a pathogen to determine if it is toxic.

Hosts, be it animals, humans or plants, have numerous pathways in them to maintain homeostasis. Consequently, due to the presence of such an enormous number of pathways in hosts, it is practically inefficient, if not impossible, to experimentally determine the effect of each of these toxins on each of the proteins involved in the host pathways. Computational models have come to our aid to save us from such laborious work. By building computational models and algorithms to mimic the actual biological scenario, we would be able to identify toxins from the proteomes (set of proteins in an organism) of such pathogens. These models and algorithms involve three crucial tasks - feature extraction, classification and pathway prediction. Thus the present thesis deals with these tasks for pathogenic toxin identification and analysis of their effect on host pathways.

Feature extraction: In this thesis, we have extracted information regarding the experimentally determined structure of toxins (primary, secondary, and tertiary). Multiple features have been extracted from the primary, secondary and tertiary structures of such toxins. Features extracted from the primary structure of toxins include nucleotide sequence profile, peptide sequence profile, solvent accessibility profile, conjoint triad descriptors and evolutionary information-based profile. Secondary structure of these toxins has provided information on the percentage composition of helices, coils and sheets. The tertiary structure of toxins has led to the generation of features, like radius of gyration, compactness, convex hull layer count, surface atom composition and packing density. Using these features, we have developed algorithms to predict toxins of pathogenic species that are not well researched with a high accuracy [351, 353, 355]. Since not all proteins have multiple polypeptide chains, we have not used quaternary structure-based features for identification of toxins.

Classification: In order to identify these toxins, machine learning methodology has been developed based on these features forming input datasets. Before these datasets could be used for classification, their imbalanced nature has been rectified using a new oversampling algorithm developed with the intention to facilitate toxin identification in an improved manner. Having obtained a balanced dataset, various machine learning methodologies with appropriate parameters have been trained to develop systems for the identification of such pathogenic toxins [351, 353, 355]. The systems developed have been made to undergo multiple testing procedures and subjected to biological validation to ensure its robustness, efficiency and accuracy in identification of toxins. These systems have been made available to facilitate further research in this domain.

Pathway Prediction: Not just in the identification, computational algorithms developed in this thesis have facilitated in understanding the effect of such toxins on host pathways. We have used the structural characteristics of metabolites to predict the effect of toxins on metabolic pathways [356]. Additionally, how toxins affect the progression of metabolic pathways has been experimented with and documented. We have also developed algorithms to study the effect of toxins on signaling pathways, by introducing a new measure to quantitatively define robustness of such pathways and how robustness gets affected by toxins [352]. We have converted these algorithms into software systems so that they are readily accessible for research and application in future.

1.4 Preliminaries of the thesis

In this section, we briefly describe the computational and mathematical concepts being used in the thesis.

1.4.1 Mapping biological problems onto graphs

A broad spectrum of biological problems can be mapped onto graphs for an effective analysis. Diseased or normal pathways can be represented in the form of networks or graphs.

In metabolic pathways, the metabolites are represented as vertices while the transformations among these metabolites are represented as edges. For signaling pathways, the proteins are represented as vertices and the interaction among these proteins are represented as edges. For example, the glycolysis pathway (Figure 1.1), an important metabolic pathway, and ERPB signaling pathway (Figure 1.2), can be represented as graphs. Considering the glycolysis pathway (Figure 1.1), compounds can be represented as vertices (circle) while the connections between these compounds, indicating edges of a graph, can represent transformations. For ERPB signaling pathway (Figure 1.2), the proteins denoted by green boxes can be represented as vertices of a graph. The edges of a graph can represent the arrows between these boxes. Similarly, representation of the metabolites and toxins (Figure 1.3) too can be accomplished by graphs. If we consider the compound represented in Figure 1.3, its every atom can be considered as a vertex, while every bond can represent an edge. Proteins in their tertiary structure can be considered as a point cloud.

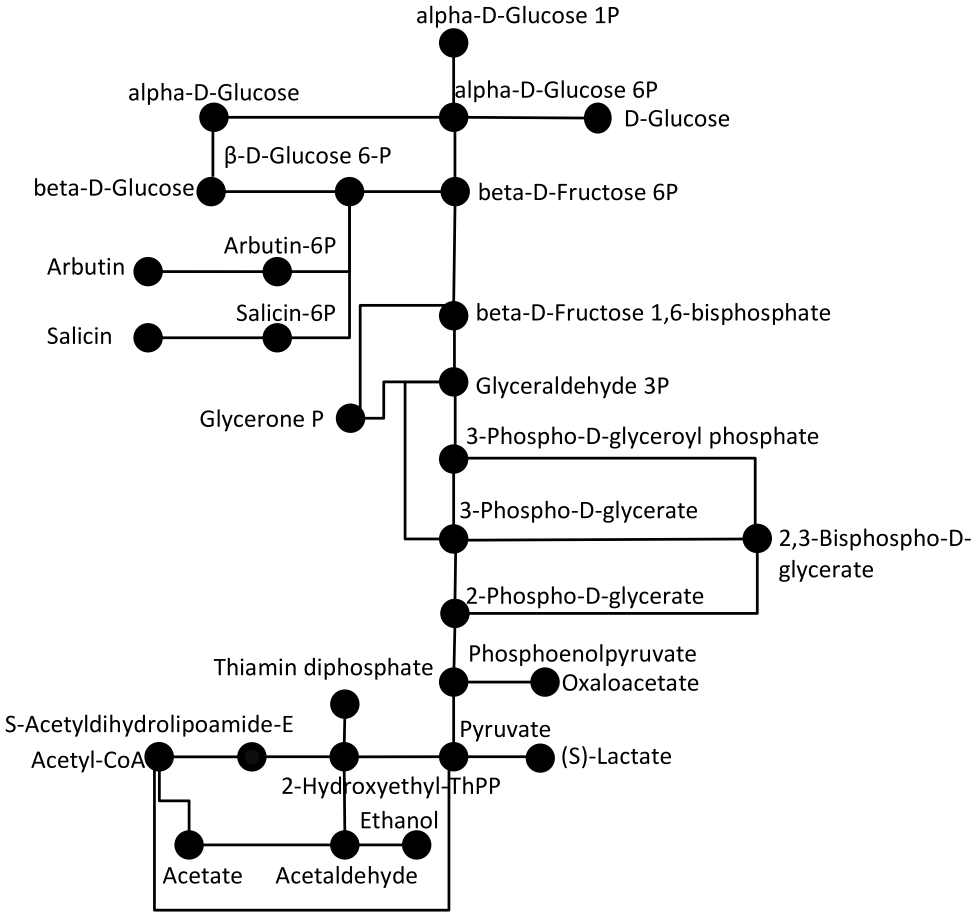


Figure 1.1: Glycolysis pathway [209]. The circles (nodes) denote metabolites, while the lines connecting these circles (edges) denote transformations.

Here, we present the formal notations and standard definitions that will be used throughout the thesis. To simplify, we use the terms "network" and "graph" synonymous. A graph

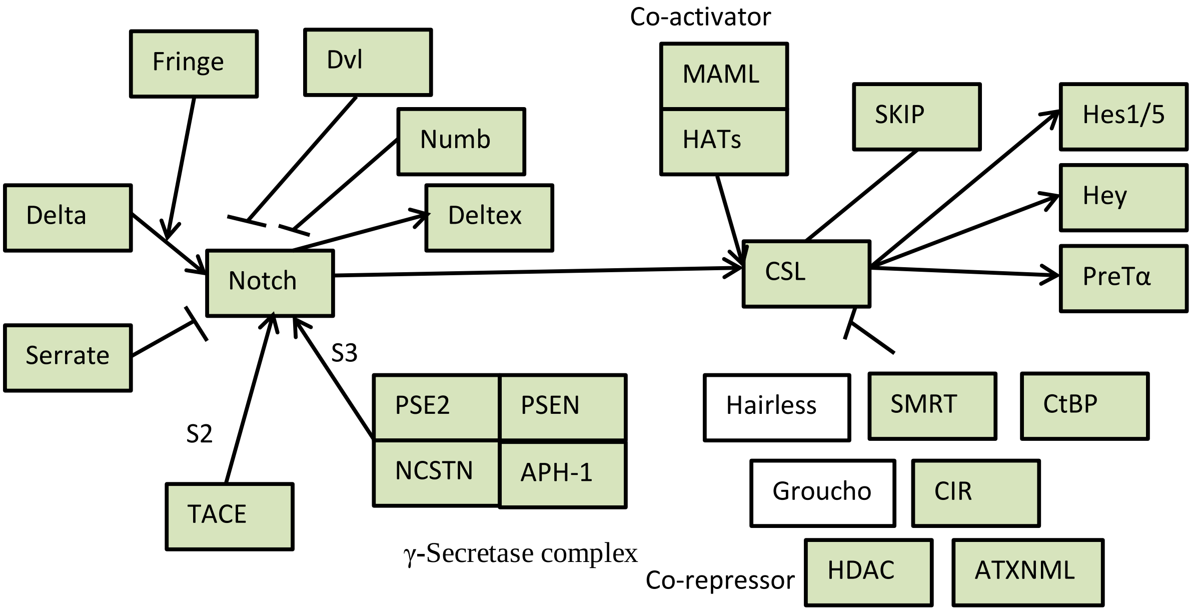
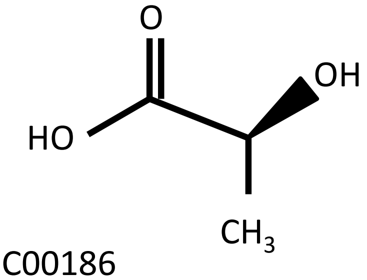


Figure 1.2: Notch signaling pathway [209]. The rectangles (nodes) denote proteins, while the lines connecting these rectangles (edges) denote interaction types.



C00186

Figure 1.3: Structure of the metabolite L-Lactate [209]. The atoms denote nodes, while the bonds between these atoms denote edges.

is represented by $G = (V, E)$, where V denotes the set of vertices, and E stands for the set of edges. A path in a graph is a sequence of edges such that every pair of subsequent edges share a common vertex. The length of a path is denoted by the number of edges it includes. A closed path starting and ending with the same vertex in a graph is defined as a cycle. A subgraph of a graph contains a subset of the vertices and edges. Further terms will be formally introduced whenever required.

1.4.2 Solving biological problems using machine learning

Machine Learning (ML) is the field of study that gives computers the capability to learn without being explicitly programmed. The basic idea behind the working of machine learning algorithms is by building a mathematical model based on sample data, also known as training data, to make predictions or decisions without being explicitly programmed to perform the task.

ML algorithms facilitate low-cost solutions to the time consuming and laborious experimental approaches for tasks such as sequence identification, determining groups of co-expressed genes, protein structure prediction, gene prediction, modeling of complex interactions in biological systems, precision medicine and text-mining [28], among others. ML algorithms learn how to combine multiple features of the input data into a more abstract set of features from which further learning can be conducted [239]. The multi-layered approach to learning patterns in the input data allows such systems to make complex predictions when the learning system is trained on large datasets. With time, size and number of available biological datasets have skyrocketed, driving bioinformatics researchers to make use of machine learning algorithms [451].

ML algorithms are classified into several broad categories. Under unsupervised learning, an ML algorithm builds a mathematical model from a set of data that contains only input and no desired output. Unsupervised learning algorithms are used to find structure in the data, like grouping or clustering of data objects. In supervised learning, an ML algorithm builds a mathematical/computational model from a set of data that contains both input and desired output [279]. Regression and classification algorithms work under supervised learning. Classification algorithms are used when the outputs are restricted to a limited set of values. Regression algorithms may have continuous outputs *i.e.*, any value within a domain. The most widely used machine learning algorithms include k-means clustering algorithm, support vector machines (SVM), linear regression, logistic regression, naive Bayes (NB) classifier, decision tree (DT) algorithm, k-nearest neighbor (kNN) classifier and artificial neural networks (ANN).

1.4.3 Preliminaries of convex hull

In mathematics, a set of points P is said to be convex if for any two points p, q in P, any point on the line segment pq belongs to P. The convex hull or convex envelope or convex closure of a set P of points in the Euclidean space is the smallest convex set that contains P [129, 368]. In order to understand convex hull in 3D Euclidean space, one must first understand the concept of convexity. The convex hull in 2D is a polygon, while in 3D, it is a polyhedron. A convex polyhedron is a special case of a polyhedron, having the additional property that it is also a convex set of points in $\mathbb{R}^3$.

The polygons forming a polyhedron (or bounding a solid polyhedron) are referred to as the faces of the polyhedron, provided that all coplanar polygons with common sides or segments of sides are treated as a single polygon, thus making a single face. The sides and vertices of the faces of a polyhedron are referred to as the edges and vertices of the polyhedron respectively. In this thesis, the convex hull in the 3D space has been derived by

Quickhull algorithm [30].

1.4.4 Performance measures

The performance of the different algorithms and classifiers developed/implemented in this thesis have been assessed by *Accuracy, Sensitivity, Specificity [442], F-score, G-mean, Receiver Operating Characteristic (ROC), Area Under Curve (AUC), Matthews Correlation Coefficient (MCC) and Cohen's κ score [374]*. They are defined below:

$$\textit{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.1}$$

$$\textit{Sensitivity (TPR)} = \frac{TP}{TP + FN} \tag{1.2}$$

$$\textit{Specificity} = \frac{TN}{TN + FP} \tag{1.3}$$

$$\textit{Precision} = \frac{TP}{TP + FP}. \tag{1.4}$$

$$\textit{FPR} = \frac{FP}{FP + TN}. \tag{1.5}$$

$$\textit{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{1.6}$$

$$\kappa = \frac{p_0 - p_e}{1 - p_e}. \tag{1.7}$$

where

$$p_0 = \frac{TP + TN}{TP + TN + FP + FN}. \tag{1.8}$$

and

$$p_e = \frac{(TN + FP)(TN + FN) + (FN + TP)(FP + TP)}{(TP + TN + FP + FN)^2}. \tag{1.9}$$

Here TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative values respectively. $Accuracy$ reflects the ability of a classifier in discriminating classes. $Sensitivity$ is the measure of the proportion of actual positives that are correctly identified as such, while $Specificity$ gives the proportion of actual negatives that are correctly identified as such. $Precision$, on the other hand, signifies the proportion of correct predictions in the positive class.

However, for an unbalanced class, $Accuracy$ is not a very reliable metric to measure the performance of a classifier. Let us consider a dataset with two classes only, where the first class contains 90% of the data (majority class), and the second constitutes the remaining 10% (minority class). If the classifier predicts every sample belonging to the first class, the accuracy will be of 90%, although this classifier is in practice useless. Getting a high accuracy for imbalanced classes is easy, without actually making useful predictions. Thus, $Accuracy$ as an evaluation measure makes sense only if the class labels are uniformly distributed [391]. The curves AUC-ROC are not sensitive to imbalanced datasets. This is because a small number of correct or incorrect predictions can result in a large change in the ROC curves. ROC is obtained by plotting false positive rate versus true positive rate. As the extent of data imbalance increases, true positive rate (TPR) will mostly remain constant since it depends on misclassifying minority class examples. In the case of an imbalanced dataset, there will be more FP. Since the majority class has more data, TN will also be high. As a result, false positive rate (FPR) remains the same. Given that both equations remain the same intuitively, it is evident that AUC-ROC is not sensitive to imbalanced datasets [391].

For a more robust performance measure for classification of imbalanced datasets containing samples from either of the two classes, we consider F-score and G-mean. F-score [51] combines *sensitivity* and *precision*, and is given by

$$\begin{aligned} F\text{-score} &= \frac{2}{\frac{1}{\textit{precision}} + \frac{1}{\textit{sensitivity}}} \\ &= \frac{2TP}{2TP + FP + FN} \end{aligned} \tag{1.10}$$

On the other hand, G-mean [135] attempts to maximize the accuracy across two classes with a good balance, and is defined as

$$\begin{aligned} G\text{-mean} &= \sqrt{\textit{sensitivity} \times \textit{specificity}} \\ &= \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{FP+TN}} \end{aligned} \tag{1.11}$$

Larger the values of F-score and G-mean, better is the classifier.

The measures mentioned above are suitable for 2-class problems. We need to know the performance measure for multiclass classification on imbalanced datasets. For multiclass classification, we have used *Accuracy, Matthew's correlation coefficient (MCC)* and *Cohen's kappa* (κ) score.

Let m_{ij} be the number of samples being in ith class, having been predicted to belong to jth class. For a b-class classification problem, c, s, t_j, p_j are defined as

- $s = \sum\limits_{i=1}^{b} \sum\limits_{j=1}^{b} m_{ij}$ is the total number of samples
- $c = \sum\limits_{i=1}^{b} m_{ii}$ is the total number of samples correctly predicted
- $t_j = \sum\limits_{i=1}^{b} m_{ji}$ is the total number of samples in class j

- $p_j = \sum_{i=1}^{b} m_{ij}$ is the number of samples predicted to be in class j

Cohen's κ score, for multiclass classification problems, is given by [386]

$$\kappa = \frac{c \times s - \sum_{j=1}^{b} p_j t_j}{(s^2 - \sum_{j=1}^{b} p_j t_j)} \tag{1.12}$$

For multiclass classification problems, *MCC* is defined as [204]

$$MCC = \frac{c \times s - \sum_{j=1}^{b} p_j t_j}{\sqrt{(s^2 - \sum_{j=1}^{b} p_j^2)(s^2 - \sum_{j=1}^{b} t_j^2)}} \tag{1.13}$$

Cohen's κ score is simple and widely used for measuring the performance of a classifier dealing with more than two classes [34, 149]. The value of κ is always less than or equal to 1, where a score less than 0 indicates a random prediction. Closer the value of κ to 1, better is the prediction. MCC takes true/false positives/negatives into account, and is generally regarded as an equal measure [204]. *MCC* values lie between -1 and +1, where the value of +1 represents a perfect prediction, 0 an average random prediction, and the value of -1 indicates an inverse prediction. Both *Cohen's* κ score and *MCC* are sensitive to imbalanced data [391].

1.5 Scope and Organization of the thesis

This thesis is a comprehensive attempt to the identification of pathogenic toxins and their effect on signal transduction and metabolic pathways of the hosts based on the principles of feature extraction, classification and pathway prediction. The present chapter deals with the concept of pathogen, host, host-pathogen interactions, and the importance of computer science in host-pathogen interactions. The second chapter gives an overview of how computer science has helped in generating knowledge in the field of host-pathogen interactions. The remaining five chapters constitute the contributory part of the thesis, followed by a concluding chapter. The content of the chapters has been outlined in Figure 1.4.

1.5.1 *Chapter 2* - A Review on Host-Pathogen Interactions: Classification and Prediction

Chapter 2, being on literature survey, describes the current research in host-pathogen interactions [354]. It covers the biological and computational aspects of host-pathogen interactions, classification of the methods by which the pathogens interact with their hosts, different machine learning techniques for prediction of various toxins and protein-protein interactions, and future scopes of this research field.

1.5.2 *Chapter 3* - PyPredT6: An Ensemble Learning-based System for Identification of Type VI Effector Proteins

Chapter 3 contributes to the development of a predictor system, called PyPredT6, for prediction of Type VI (T6) effector proteins, by utilizing information based on their primary and secondary structures [353]. Prediction of T6 effector proteins is a new challenge since the discovery of the T6 Secretion System. A total of 873 unique features have been extracted from the peptide and nucleotide sequences of the experimentally verified effector proteins. Based on these features and using ensemble learning, we have performed *in silico* prediction of T6 effector proteins in *Vibrio cholerae* and *Yersinia pestis* to demonstrate the effectiveness of PyPredT6. PyPredT6 has been seen to provide better prediction in comparison with other T6 effector protein predictors, with a reported accuracy of 92.15%. While analyzing the feature set, a considerable difference has been noticed in the distribution of α-helices and β-sheets in effectors with respect to the non-effectors ($p < 0.05$). The implementation of the method PyPredT6 is available at `http://projectphd.droppages.com/PyPredT6.html`.

1.5.3 *Chapter 4* - Cluster Quality-based Non-Reductional (CQNR) Oversampling Technique and Effector Protein Predictor Based on 3D Structure (EPP3D) of Proteins

Chapter 4 deals with identification of effector proteins based on their 3D structure, incorporating a novel oversampling algorithm. Currently, no mechanism to identify the effector proteins based on their 3D structure has been reported in the literature. In order to identify effector proteins, extraction of features from their 3D structure is crucial. However, effector protein datasets are highly imbalanced. State-of-the-art oversampling algorithms are incapable of dealing with such datasets. They usually eliminate samples as noise and do not ensure generation of synthetic samples strictly in the vicinity of the minority class samples.

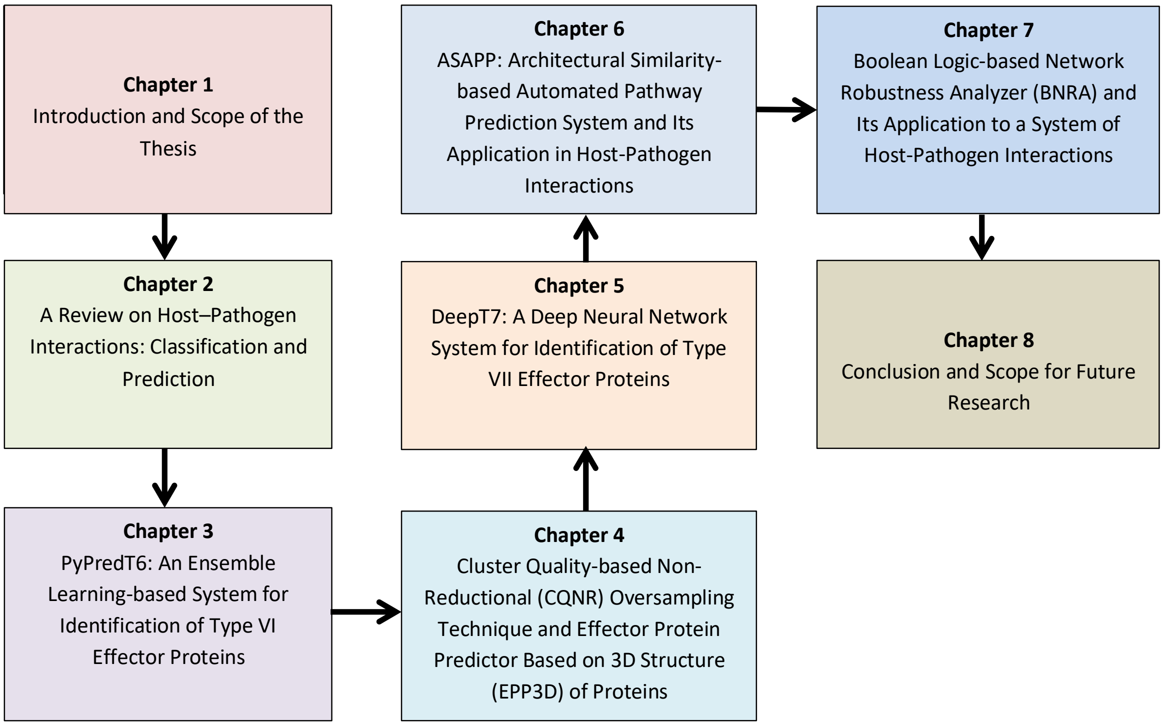

Chapter 1
Introduction and Scope of the Thesis
Chapter 2
A Review on Host–Pathogen Interactions: Classification and Prediction
Chapter 3
PyPredT6: An Ensemble Learning-based System for Identification of Type VI Effector Proteins
Chapter 4
Cluster Quality-based Non-Reductional (CQNR) Oversampling Technique and Effector Protein Predictor Based on 3D Structure (EPP3D) of Proteins
Chapter 5
DeepT7: A Deep Neural Network System for Identification of Type VII Effector Proteins
Chapter 6
ASAPP: Architectural Similarity-based Automated Pathway Prediction System and Its Application in Host-Pathogen Interactions
Chapter 7
Boolean Logic-based Network Robustness Analyzer (BNRA) and Its Application to a System of Host-Pathogen Interactions
Chapter 8
Conclusion and Scope for Future Research

Figure 1.4: Outline of the thesis.

In effector protein datasets, deletion of any samples as noise would lead to loss of crucial information. Furthermore, generation of synthetic samples of the minority class in the vicinity of majority class samples would lead to an inept classifier.

In this chapter, we develop an algorithm, called Cluster Quality-based Non-Reductional (CQNR), for the purpose of oversampling minority classes. Its novelty lies in generating new samples proportional to the distribution of samples of the minority classes, without eliminating any sample as noise. Utilizing CQNR, we develop a novel Effector Protein Predictor based on the 3D (EPP3D) structure of proteins. EPP3D is trained on a feature set, comprising features, *viz.*, convex hull layer count, surface atom composition, radius of gyration, packing density and compactness, derived from the 3D structure of the experimentally verified effector proteins. F-score and G-mean demonstrate that CQNR has outperformed some well-established oversampling methods by approximately 3–5% on five benchmark datasets and three other synthetically generated highly imbalanced datasets. Likewise, for classification of pathogenic effector proteins, a significant improvement of 7–9% in accuracy has been noticed on the application of CQNR followed by EPP3D with respect to other oversampling algorithms. Moreover, EPP3D has exhibited an improvement of 2–4% in classifying effector proteins based on their 3D structure compared to the classification of effector proteins based on their amino acid sequences. The software for CQNR and EPP3D are available at `http://projectphd.droppages.com/CQNR.html`.

1.5.4 *Chapter 5* - DeepT7: A Deep Neural Network System for Identification of Type VII Effector Proteins

Following the successful application of the techniques to uniquely identify Type VI proteins in Chapter 3, and using 3D structure to identify the effector proteins in Chapter 4, we extend these investigations towards identifying Type VII proteins. An important group of pathogens that have been known to secrete virulence factors are organisms harboring the Type VII Secretion System (T7SS). Prediction of T7 effector proteins has become crucial since the discovery of T7 secretion system. In Chapter 5, we develop a Deep Neural Network system, called DeepT7, to predict T7 effector proteins. The nucleotide and peptide sequences of experimentally verified effector proteins have been considered for constructing a set of 1727 features. The feature set captures various aspects of effector proteins, which include their physicochemical properties, primary and secondary structure-based information, and evolutionary information-based properties. The effectiveness of DeepT7 has been demonstrated on the proteomes of two organisms, *Mycobacterium bovis* and *Streptococcus pneumoniae*, known to possess T7SS, for predicting their T7 effector proteins. The outcome of DeepT7 has been biologically validated. DeepT7 has identified T7 effectors with an *accuracy* of 91.50%, *sensitivity* of 91.10%, *specificity* of 99.14%, F-score of 0.6721, G-mean of 0.9504, Cohen's κ score of 0.6467 and MCC of 0.7480. DeepT7 is available at `http://projectphd.droppages.com/DeepT7.html`.

1.5.5 *Chapter 6* - ASAPP: Architectural Similarity-based Automated Pathway Prediction System and Its Application in Host-Pathogen Interactions

The previous chapters are dedicated to the identification of perturbing agents. However, a holistic study of this would remain incomplete without an analysis of the effect of these toxins on host pathways, which we have attempted to do in Chapters 6 and 7. Chapter 6 is dedicated to the design of a novel generalized algorithm, called Architectural Similarity-based Automated Pathway Prediction (ASAPP), which is used to predict metabolic pathways based on the structural resemblance of the metabolites. The significance of metabolic pathway prediction is to envision the viable unknown transformations that can occur provided the appropriate enzymes are present. It can facilitate the prediction of the consequences of host-pathogen interactions. ASAPP takes two-dimensional structure and molecular weight of metabolites as input, and generates a list of probable transformations, without the knowledge of any externally established reactions, with an accuracy of 85.09%. ASAPP has also been applied to predict the outcome of pathogen liberated toxins on the carbohydrate and lipid pathways of the hosts. We have analyzed the disruption of host pathways in the presence of toxins, and have found that some metabolites in glycolysis and TCA cycle have a high chance of being the breakpoints in the pathway. The algorithm ASAPP is available at http://asapp.droppages.com/.

1.5.6 *Chapter 7* - Boolean Logic-based Network Robustness Analyzer (BNRA) and Its Application to a System of Host-Pathogen Interactions

Effect of toxins on metabolic pathways has been studied in the last chapter. In this chapter we move our focus to studying the effect of toxins on signal transduction pathways. Computational modeling of signal transduction pathways promises to uncover the working mechanism governing such networks. It paves the way for studying the effect of pathogenic substances and assists in the development of drugs that would work on infected networks. Perturbation of biological networks leads to diseases. More robust a network, less prone it is to get perturbed. Understanding these perturbations within the context of biological networks is one of the significant challenges in systems biology. It has been seen that computational models reveal logical interrelations between Boolean networks and signal transduction pathways.

In this light and in order to get a better insight into these networks, we design an algorithm, called Boolean logic-based Network Robustness Analyzer (BNRA), in Chapter 7. BNRA models biological pathways in the form of undirected networks. The algorithm computes the robustness scores of both perturbed and unperturbed networks. It defines a quantitative measure to reflect the robustness of a network, before and after perturbation. BNRA has been applied to 221 pathways belonging to 26 categories, including human disease networks. Among these 221 pathways, four of them, viz., mRNA surveillance pathway, transcriptional misregulation in cancer, hypertrophic cardiomyopathy (HCM) and synaptic vesicle cycle, have been found to be the most robust among all the pathways considered here. An analysis of BNRA on disease pathways, including the recently discovered COVID-19 pathway, has also been provided. BNRA is available for download at http://projectphd.droppages.com/BNRA.html.

1.5.7 *Chapter 8* - Conclusions and Scope for Future Research

Finally, in Chapter 8, we provide concluding remarks on the algorithms designed as well as the results generated by them. We provide an insight into the limitations of each of the developed algorithms. We also give, in this chapter, a brief description of the future scope of the thesis.

Chapter 2

A Review on Host–Pathogen Interactions: Classification and Prediction [354]

2.1 Introduction

As discussed in Chapter 1, the term ‘host-pathogen interaction’ refers to the ways in which a pathogen (virus, bacteria, prion, fungus and viroid) influences activities of its hosts, and vice-versa. Pathogens adapt to changes, and find alternative ways to survive and infect a host. Questions like how the pathogens function, how their entry point into the host is facilitated through the biological barriers and how they survive inside a host that is often under treatment or immunized for the same pathogen, can be answered by exploring host-pathogen interactions. Host-pathogen interactions can be described on the population level (virus infections in a human population), on the organismal level (pathogens infecting host), or on the molecular level (pathogen protein binding to a receptor on human cell). However, before stepping into methodological details of host-pathogen interaction processes, a brief glimpse into history of this research field is included here to sum up the how(s) and why(s) of recent advancements of this field.

Some of the earliest investigations in the domain of host-pathogen interactions are: i) study of host-pathogen interaction in mouse typhoid caused by *Salmonella typhimurium* [448], ii) genetic study of physiology of parasitism of the corn rust pathogen *Puccinia sorghi* [114], iii) a correlation study of α-galactosidase production and host-pathogen interaction between *Phaseolus vulgaris* and *Colletotrichum lindemuthianurn* [134], iv) study of ultra-structural aspects of a host-pathogen relationship of a deuteromycetes fungus, *Pyrenochaeta terrestris* with 2 *Allium cepa* (onion) varieties with the help of electron microscopy [188], v) fine structure study of principal infection procedure during infection of Barley by *Erysiphe* *graminis* [128], vi) a study on proteins which obstructs the action of the polygalacturonases (polygalaicturonide hydrolases, EC 3.2.1.15) released by the fungal plant pathogens *Fusarium oxysporum*, *Colletotrichum lindemuthianum*, and *Sclerotium rolfsii*. These proteins are extracted from the cell walls of red kidney bean hypocotyls, tomato stems and suspension-cultured sycamore cells [8],vii) a study on proteins secreted by plant pathogens which impedes enzymes of the host having the ability to attack the pathogen. The study is conducted on a interaction system of a fungal pathogen (*Colletotrichum lindemuthianum*) and its host, the French bean (*Phaseolus vulgaris*) [9], viii) a study on a single plant protein that efficiently hinders endopolygalacturonases secreted by *Aspergillus niger* and *Colletotrichum lindemuthianum* [143], ix) a molecular basis study to showcase mutation of *Xanthomonas campestris* to overcome resistance in pepper (*Capsicum annuum*) [214], x) a study on stress and immunological response in host-pathogen interactions [289].

Some recent research works have focused on

- the basic notion of virulence and pathogenicity which defines and suggests a classification system for microbial pathogens based on their capacity to cause damage as a consequence of the host's immune response [66],
- model organisms for host-pathogen interactions, i.e., *C. elegans* [236], *D. melanogaster* [290,404] and zebrafish [171,396] among others,
- molecular cross-talk of host-pathogen interactions where Type III secretion system is mentioned [340],
- novel studies involving epigenetics[1] [157], metallobiology [48], quantitative temporal viromics[2] [419], heterogeneity in same host tissue [56], and computational systems biology [123] of host-pathogen interactions.

All these investigations indirectly show us the trend of development of the host-pathogen interactions research field. The field has started with sporadic research works dedicated towards pathogens and their interactions with their hosts. The earliest research has been done on host-pathogen interactions with respect to environmental factors, like light, temperature, season, and pathogen/host population among others. Later some organisms, like *C. elegans* and *D. melanogaster* have been found as model organisms to study the pathogen behavior of other complex hosts (human beings) due to their easy body plan, known genome structure and short life cycle. Gradually, certain proteins and then protein clusters have been marked for taking part in host-pathogen interactions. Moreover, definite classification has been found for the mechanism of host-pathogen interactions at the advent of recent developments in

[1] a procedure through which genotypes give rise to phenotypes during development due to changes in underlying DNA sequence(s), i.e., histone modifications, DNA methylation, DNA silencing via noncoding RNAs and chromatin remodeling proteins.

[2] temporal alterations in host and viral proteins throughout the course of a productive infection

imaging and molecular biology techniques.

Moreover, some research works have defined and gave direction to the host-pathogen interactions research field. Discovery of distinct secretion systems [109,147,234,317,319,421] has provided the basic background of host-pathogen interaction research. The concerned studies have spanned from genome locus [234] to biochemical and genetic evidence [286]. With discovery of protein-protein interaction (PPI) prediction methods [44], the chance of finding host-pathogen protein pairs and their interactions has become more prominent and such studies have given a different direction to the research field. Methods have been developed for machine learning-based *in silico* prediction of secretion system associated proteins [17]. There are also a couple of newly proposed methods [180, 275] which provide new glimmer of hope to the research field in controlling pathogenesis in a host as described below.

- Secretion systems Type I [421], Type II [109], Type III [147] and Type V [317] have been discovered in 1980s, which have defined the base for host-pathogen interaction research.

- Kuldau *et al.* [234] have predicted 11 ORFs from virB locus in 1990. Based on hydropathy plot they have analyzed that nine of these encode proteins which may interact with membranes and may form a membrane pore or channel to mediate exit of the T-DNA copy. This is the first indirect indication of a distinct secretion system, later known as Type IV Secretion system (T4SS).

- Pukatzki *et al.* have functionally defined T6SS in 2006 [319].

- Mougous *et al.* in 2006 have provided biochemical and genetic evidence that a virulence-associated genetic locus of *P. aeruginosa*, termed as HSI-I, encodes a protein secretion apparatus (T6SS) [286].

- Machine learning-based prediction of PPIs have been done by Bock *et al.* in 2001 [44]. They have used Support Vector Machine (SVM) to train and predict interactions based on primary structure and related physicochemical properties. This work has provided a shift in research direction from genes to their protein counter parts and their nature of interaction.

- First ever machine learning-based prediction of Type III secretion system associated proteins have been done by Arnold *et al.* in 2009 by analyzing the amino acid composition and secondary structure composition of a few experimentally verified effector proteins at N-terminal [17].

- A few new studies and methods have proposed new avenues of future host-pathogen interaction research, i.e., a new way of studying host-pathogen interaction by dendritic cell subtypes [275] and chemoproteomic profiling of host and pathogen enzymes for finding candidates (proteases) to disrupt pathogenic mechanisms which often have boosted the host's defense mechanisms directly or indirectly [180].

The present literature survey tries to encompass the *in silico* prediction of host-pathogen interactions by machine learning and the related methodologies. It has been organized into dedicated sections of classification of host-pathogen interactions, availability of host-pathogen interaction data, prediction of host-pathogen interaction domains, image processing-based research techniques, and conclusive remarks. There are several substrates and pathways whereby pathogens can invade a host. The human body has its own natural defense mechanism against some of the common pathogens in the form of the immune system that acts against these pathogens. Pathogens have the capability to adhere to host tissues, to evade host defenses, and to invade host cells. However, deeper understanding has revealed that each pathogen has their own variation of these themes [336]. Host-pathogen interactions take place between a host and a pathogen through the protein(s) and gene(s), and by disrupting normal functioning of pathway(s), forming biofilm(s), inhibiting macrophage activity and by other methods. This survey has briefly discussed about the various probable factors which directly or indirectly contribute to host-pathogen interactions. Pathogens can either attack a host in gene level by emitting RNA, or they can release proteins which would lead to pathogenicity or they can inhibit the mechanism of macrophage. Some pathogens utilize the components of a host system to survive in the host. These components are called host factors. In a few cases, some factors of a pathogen can initiate the autophagy mechanism which acts in favor of the host. The classification of the host-pathogen interactions is based on traditional pathogen invasion into host.

The survey starts with categorization (Figure 2.1) of pathogens, and makes a comprehensive list of diseases caused by them. The following section discusses classification of host-pathogen interactions based on different biology-based reasoning. Following this, is the description of the widely used *in silico* prediction methods in the domain of host-pathogen interactions. Moreover, an extensive list of the online repositories has been furnished. The survey concludes with a brief discussion that includes the merits and demerits of this research field in general, a few scopes for future research and concluding remarks.

2.2 Classification of Host-Pathogen Interactions

This section describes briefly basic biology concerning host-pathogen interactions. The components of a host-pathogen interaction can be broadly classified into four stages, *i.e.*, invasion of host through primary barriers, evasion of host defenses by pathogens, pathogen replication in host and a host's immunological capability to control/eliminate the pathogen. A pathogen

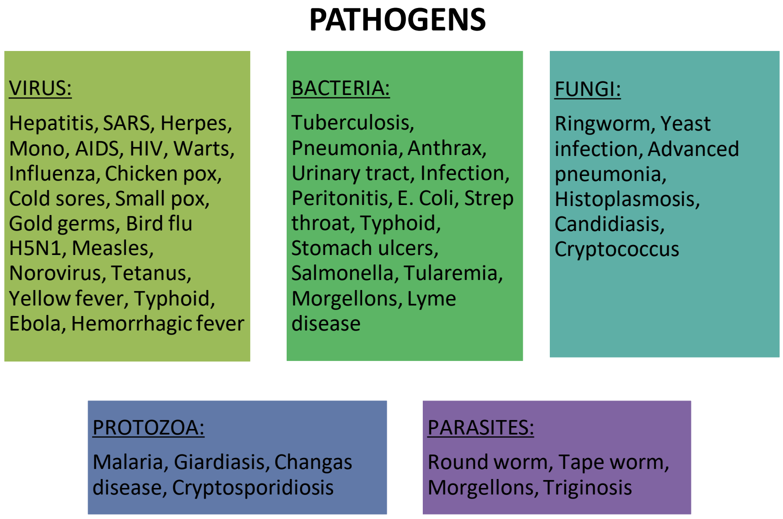

PATHOGENS
VIRUS:
Hepatitis, SARS, Herpes, Mono, AIDS, HIV, Warts, Influenza, Chicken pox, Cold sores, Small pox, Gold germs, Bird flu H5N1, Measles, Norovirus, Tetanus, Yellow fever, Typhoid, Ebola, Hemorrhagic fever
BACTERIA:
Tuberculosis, Pneumonia, Anthrax, Urinary tract, Infection, Peritonitis, E. Coli, Strep throat, Typhoid, Stomach ulcers, Salmonella, Tularemia, Morgellons, Lyme disease
FUNGI:
Ringworm, Yeast infection, Advanced pneumonia, Histoplasmosis, Candidiasis, Cryptococcus
PROTOZOA:
Malaria, Giardiasis, Changas disease, Cryptosporidiosis
PARASITES:
Round worm, Tape worm, Morgellons, Triginosis

Figure 2.1: Classification of some common pathogens and the list of diseases caused by them

can invade a host only after breaching the primary host defenses. Pathogens contain virulence factors which promote and cause disease. The greater the virulence, the more likely the disease will occur. The entire process of host-pathogen interactions has been classified according to these stages. A summary of the methods discussed in this survey has been diagrammatically represented in Figure 2.2. However, *in silico* prediction methods used for

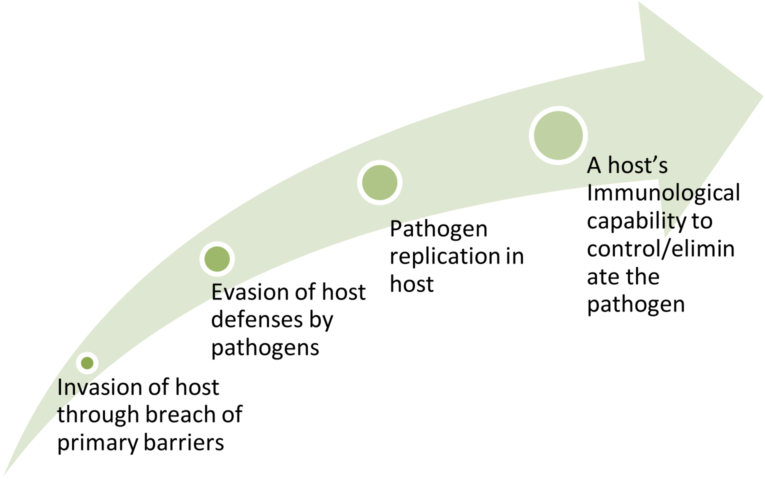

Invasion of host through breach of primary barriers
Evasion of host defenses by pathogens
Pathogen replication in host
A host's Immunological capability to control/elimin ate the pathogen

Figure 2.2: Classification of Host-Pathogen Interactions

detection of such interactions have been described in the Section 2.3. The stages mentioned below are overlapping in nature. They do not have a clear boundary between them. The *in silico* prediction methods described later cannot be uniquely associated to only one of the stages. Their applicability spans over many or all the stages of host pathogen interactions.

2.2.1 Invasion of host through breach of primary barriers

One of the main ways in which pathogens invade the host is via protein secretion. Pathogens, particularly the gram-negative bacteria, which cause pathogenesis in host, consist of secretion systems. These secretion systems release proteins, called effectors, into the body of the host when they come in contact with the host. There are at least six specialized secretion systems in gram-negative bacteria. Type I, Type II, Type III, Type IV, Type V and Type VI are the prominent ones based on their mechanisms of host infection. Details of these mechanisms can be obtained from Costa *et al.* [93]. Numerous secreted proteins are crucial in bacterial pathogenesis. A few of them has been described here, *i.e.*, toxins, urease and multivalent adhesion molecule.

In contrast to gram-negative bacteria, gram-positive bacteria are generally regarded as being simpler in structure because they lack a second membrane; consequently, secretory proteins of gram-positive bacteria only need to traverse the cytoplasmic membrane and peptidoglycan layer to enter the extracellular environment. However, recent studies have provided evidence that there is an alternative protein-secretion system in gram-positive bacteria. Perhaps unsurprisingly, this specialized secretion system has been identified in *Mycobacterium tuberculosis*, a gram-positive bacterium with a highly complex cell envelope.

Apart from effector proteins, metabolic compounds known as toxins too harm the host in many ways. Toxins are substances released by pathogens that are poisonous to humans. Most toxins that cause problems in humans come from germs such as bacteria. Toxins are capable of causing disease on contact with or absorption by body tissues interacting with biological macromolecules such as enzymes or cellular receptors. These toxins, once in the body of the host, intervene with the normal functioning of the metabolism of host. Minimized toxin expression in a pathogen have a lesser effect on the host at the time of attack than that with higher toxin expression. The molecules that are secreted by gram-negative pathogens, lead to damage of the host cells. The vesicle released from the enclosure of the growing bacteria, serves as containers for the proteins and lipids of the gram-negative bacteria. It suggests the importance of vesicle mediated toxin delivery for the onset of infection in the host.

Effectors proteins are secreted by pathogenic bacteria for their entry into host and are crucial for virulence. These proteins assist pathogens in invading host tissue, suppressing the host's immune system, and in its survival within the host. For example, in *Yersinia pestis* (the causative agent of plague), loss of the T3SS has rendered the bacteria completely avirulent [269]. Naive Bayes classifier and support vector machine have already been applied to detect effector proteins of T3SS [17,412]. More details regarding the methodology is given in the Section 2.3.

Urease (an enzyme) plays an important role in *Mtb*-host interaction [86]. Urease is present in many species of *mycobacterium*, and its presence/absence is frequently used in the speciation of *mycobacterium*. Urease has been considered to be a virulence factor for several pathogenic microorganisms. Generation of ammonia by urease of urinary pathogens, such as *P. mirabilis*, have contributed to its pathogenesis due to its toxicity to renal epithelium, participation in complement inactivation and promotion of urinary stone formation [52]. Urease of *H. pylori* alkalizes the bacterial micro-environment in the stomach and is toxic to stomach epithelium [371]. In the case of *Mtb*, urea is readily available to the bacteria in both its intracellular and extracellular locations within the host.

Multivalent Adhesion Molecule (MAM) is responsible for establishing high affinity binding to host cells during early stages of infection [228]. MAM7 connects to a host via protein-lipid (phosphatidic acid) and protein-protein (fibronectin) interactions. MAM7 has been found on the outer membrane of the gram-negative pathogens which contributes to its virulence.

2.2.2 Evasion of host defenses by pathogens

In order to survive inside the host, pathogens need to avoid host defense mechanisms. *Mycobacterium tuberculosis (Mtb)* showcases that it actively transcribes a number of genes involved in fortification and evasion from a host system [321]. Assessment of the genome of 58 strains of *Staphylococcus aureus* reveals that all the immune evasive proteins are present in all the strains but not all the surface proteins [270]. Remarkably, four strains have surface and immune evasion genes similar to human strain. On the other hand, the putative targets of these proteins vary in different hosts, which propose that these proteins are not crucial for virulence. Signaling for anti-inflammation by glycolipids and host system interaction may be considered as a method of *Mycobacteria* to evade the host or may be playing a vital role in preventing extreme inflammatory response [398].

Pathogens often affect the essential pathways of their hosts with the aim to evade host defenses. NF-κB family of transcription factors help in the development of the APC (Antigen Presenting Cell) and the lymphocyte [389]. Once the host is compromised, NF-κB pathway gets activated. HIV-1 mostly depends on its host for survival as it has a few genes of its own. An integrated study of HIV-1 and human signal transduction pathways have been carried out to infer that most of these pathways may get effected by HIV virus during its life cycle [27]. It has assessed and analyzed all possible paths (perturbed and unperturbed) starting from one protein (start point) terminating into another (end point).

Human proteins potentially targeted by Epstein-Barr virus (EBV), tend to be hubs in the human interactome. It is consistent with the hypothesis that hub protein targeting is an effective mechanism for viruses to convert pathways for their use [61]. Bacterial and viral pathogens are more inclined to interact with hub proteins, and the proteins that are central to multiple pathways in the network [126]. Certain cellular mechanisms, like cell cycle regulation and nuclear transport participate in these interactions with a different set of pathogens. A study has identified 3073 human-*B. anthracis*, 1383 human-*F. tularensis* and 4059 human-*Y. pestis* PPIs (Protein-Protein-Interactions) [127]. As suggested by Ranet *et al.* [126], these PPIs have occurred among those hub and bottleneck proteins. The extracellular hydrolytic enzymes, especially the aspartyl proteinases (Saps) secreted by *C. albicans*, are major factors of its pathogenicity [291]. Proteins Chaperon 60 and 60.1 have a higher impact on activation of the cytokines than the protein Chaperon 60.2 [250]. In *Staphylococcus aureus*, proteins EsxA and EsxB act as virulent factors to enforce pathogenesis [60]. Mutants that do not secrete these proteins have been observed for failing to enforce strong pathogenesis. Among two closely related families of proteins, PE and PE_PGRS, PE_PGRS of *Mtb* activates a considerable humoral immune response but not PE [108]. Further study suggests that unlike PE, certain PE_PGRS genes are expressed during infection and antibody response. In case of Enterovirus, 71 genes out of 699 get differentially expressed significantly during infection [262]. Lack of the flagella gene in *Salmonella typhimurium* contributes to its virulence. Addition of flagella gene increases the cytotoxicity. However, it does not increase the production of IL-6 (InterLeukin-6) [301].

One of the crucial host defenses is the macrophage. Hence macrophage inhibition is another factor using which the pathogen evades the host immune mechanism. Macrophage activation happens due to multiple components, i.e., gene(s) encoding receptor(s), signal transduction molecule(s), transcription factor(s) and bacterial component(s) that activate toll like receptor(s) (lipopolysacharide, muramyl dipeptide, lipoteichoic acid and heat shock proteins) [293] among others. Pathogens attempt to survive in the host by preventing macrophages to act on them. It has been found that pathogens disrupt the enzymatic activity in activated macrophages by disrupting the actin filament network [163].

It has been identified that falsatin is an endogenous protease inhibitor of *Plasmodium falciparum*. Analysis of inhibition of normal functionality of macrophages to engulf pathogens and ingest killed parasites due to the functioning of ornithine decarboxylase, has been done by Nairz *et al.* [216]. Due to pathogen specific responses, interleuken-12 production is inhibited for *Mtb*, hence allowing the host to fight against the pathogen. It has been found that 26 to 37 proteins of HIV-1 are associated with MDM (monocyte derived macrophages) derived from HIV [82]. Inhibition by *Mtb* can be avoided with the help of IFN-γ and transfection of LRG-47 [170]. It has been found that *Mtb* residing in macrophage, switches to anaerobic growth [350] to evade host defense for a longer period of time.

The crosstalk of host-pathogen interactions is often governed by miRNAs [155,346,347]. The small RNAs, like siRNAs and shRNAs also play a vital role in host-pathogen interactions. Konig *et al.* [227] have studied the association of siRNAs with host-pathogen interactions. They have explored it by combining genome wide siRNA analysis along with the knowledge from human interactome database. Pathogens have Short Linear Motifs (SLiM) that have high similarity with host SLiMs. Motif mimicry is used by pathogens to rewire host signaling pathways by co-opting SLiM-mediated protein interactions to affect the host systems [403].

Pneumolysin (an enzyme) is a key virulence factor [267]. It activates multiple genes and signal transduction pathways in eukaryotes. Cytolytic effect of pneumolysin contributes to lung injury and neural damage. It sometimes induces apoptosis in neurons and other cells. It can also trigger host mediated apoptosis in macrophages, thus magnifying extermination of pathogens. Pneumolysin has a both way balancing effect on the host.

2.2.3 Pathogen replication in host

For surviving inside a host, pathogens have multiple ways to facilitate their growth by speedy replication. First of all, they need a few genes and proteins to survive effectively in the host, while a lot more genes and proteins are required for their survival outside the host. A study on the metabolic network of the pathogen, *Salmonella typhimurium*, has revealed 1083 genes catalyzing 1087 metabolic and transport reactions. This suggests that a minimal set of potent metabolic pathways within *Salmonella typhimurium*, is required for its favorable replication of *Salmonella typhimurium* within the host [322]. Erythrocytic malaria parasite needs proteases for a number of its cellular processes [308] in order to survive in the host.

Pathogens have evolved strategies to promote their survival by performing hijacking of the host cells they infect. Viruses implant their DNA sequence into the normal sequence of these hosts in the hope of their better survival [328] inside the hosts. A genome of the strain of *Mtb*, H37Rv, made up of 4000 genes comprising 4,411,529 base pairs, have a high guanine and cytosine (GC) content [87]. In this genome, 194 genes are required for the growth of *Mtb* [344]. A large number of these genes is unique to mycobacteria and its closely related species. It leads to the fact that the mechanism of infection of *Mtb* is different from other pathogenic species.

Some pathogens even respond to more than one micro-environment for their replication and survival. The genes responsible for secretion in *Mycobacterium* (Snm) protein secretion in a mutation of *Mtb*, which is *Mycobacterium smegmatis*, are homologs of their *Mtb* counterpart [91]. It suggests that some strains may have similar secretion mechanism. Four essential gene products (Sm3866, Sm3869, Sm3882c, and Sm3883c) are needed for Snm secretion. *Mtb* exists in various metabolic states. This fact indicates that it may be responsive to more than one micro-environment [141].

The genome of *Mycobacterium tuberculosis* possesses a large family of Ser/Thr protein kinases (STPKs). STPKs have been found to play an important role in cell division and cell envelope biosynthesis [283]. The outer membrane of the bacteria facilitates the interaction between a host and a pathogen [233]. *C. albicans* have the capability to colonize and infect majority of the tissues of human host, which indicates that it can have functionally distinct proteinases (enzymes performing proteolysis) so as to have enough flexibility to multiply and survive in the host.

Sometimes a host itself unknowingly facilitates/inhibits the survival of its pathogens. These facilities are referred to as the host factors. These factors help in pathogen replication, transcription, integration, growth, 198 propagation, pathogen entry, and host-pathogen interactions among others. A set of 295 cellular co-factors (of host) are essential for replication of influenza virus in the early stage [226]. Among these co-factors, 181 are highly significant in host-pathogen interactions, 219 help in efficient influenza virus growth, 23 have role in vital entry and 10 are required for post entry steps of virus replication. Small molecule inhibitors of multiple factors, including vATPase and CAMK2B, go against influenza virus replication. A set of 116 Dengue Virus Host Factors (DVHF) are needed for the propagation of DENV-2 (dengue virus type 2) [357]. Among 82 human homologs of dipteran DVHF, 42 have been identified to be human DVHF. A set of 311 host factors have been found to be responsible for the growth of HIV-1 [455]. Considering HIV dependency factors obtained previously in [50] [455], it is observed that the cardinality of the set of intersection is 311 host factors. Six newly identified host factors are AKT1, PRKAA1, CD97, NEIL3, BMP2k and SERPINB6 [455]. A set of 250 such factors in HIV has been identified [50]. Rab6 and Vps53 play role in viral entry, and TNPO3 is important for viral integration and Med28 for viral transcription. HDF genes show a stronger presence in the immune cell, thus allowing the viruses to evolve in the host cells which perform the life cycle functions needed for them to survive. A set of 213 host factors and 11 HIV encoded proteins have been found responsible for HIV-1 replication [50]. Among them, a few proteins help in regulation of ubiquitin conjugation, DNA damage response, proteolysis and RNA splicing. Forty new factors play a vital role in the process of initiation and/or kinetics of DNA synthesis. Fifteen proteins with different functions have been found to play an significant role in nuclear import or viral DNA integration.

Pathogens, like *M. laprae*, cannot survive independently. Hence, they convert the glial cells of a host into progenitor cells using which it can survive and spread infection inside the host [187]. It alters the genetic structure of the adult Schwann cells to form the progenitor cells. However, it is still unknown how long *M. laprae* can survive in the de-differentiated Schwann cells as they will eventually differentiate back into adult Schwann cells.

Often apoptosis of host factors has been found to be involved in bacterial growth and sustenance inside host [457]. Apoptosis contributes to the processes of host cell deletion method, triggering of inflammation and defense mechanism. Apoptosis by the pathogen *Bordetella pertussis* allows the pathogen to survive in the introductory stages of infection.

After the pathogen has successfully colonized the tissue of the host, it stops producing the toxin adenylate cyclase hemolysin.

Biofilm formation plays a major role in host-pathogen interactions. This is a mechanism of pathogens by which they form a biofilm for their survival in the host, often utilizing degraded host proteins *Leucobacter chromiireducens* subsp. solipictus strain TAN 31504 forms biofilm. Exposure to TAN 31504 leads to change in a few innate immunity related genes in *C. elegans* [288]. Esp (a serine protease secreted by *S. epidermidis*) degrades 75 proteins of *Staphylococcus aureus* by proteolytic activity, which include 11 proteins essential for the formation of biofilm [382]. Esp also degrades several human receptor proteins involved in colonization and infection by the pathogen for the benefit of the host.

2.2.4 Immunological capability of a host to control/eliminate the pathogen

In order to prevent occurrence of infection/disease, the host body launches immune response with respect to the pathogenic invasion, *i.e.*, high expression of certain genes [385], autophagy [366, 402], role of dendritic cells [275, 330], glycoconjugates [281, 283] and iron [116, 292] in activation/alteration of host immune system.

Host genes play an important role in its immune response. Mutated β-catenin homolog bar 1 or homeobox gene egl-5 of *C. elegans*, has resulted in defective response and hypersensitivity to *Staphylococcus aureus* [197]. Bar-1 and the fgl-5 genes function parallel to the immune response pathway taken up by *C. elegans*. Over expression of egl-5 resulted in modification of NF-κB dependent TLR2 (Toll-like receptor 2) signaling in epithelial cells suggesting the role played by these two genes in immune defense of a host. Pro-16 in E-cadherin is responsible for host specificity towards the human pathogen *Listeria monocytogenes* [242]. E-cadherin of mouse, which is 85% similar to E-cadherin of human, denotes the entry of bacterial pathogen, *Listeria monocytogenes*, by not allowing E-cadherin to interact with bacterial surface protein internalin. If Proline (Pro) in the position 16 of amino acid in human is replaced by Glutamic acid (Glu) then interaction with internalin is disabled. However in mouse, if Glu is substituted by Pro then interaction with internalin is enabled. On *Mtb* interaction with mice, a group of 67 genes in an immuno-competent host has showed a high level of expression than the immuno-deficient host often in 21 days. This shows that 67 genes are responsible for immunity of mice (host) [385].

Autophagy is another mechanism of host's defense against pathogen. Autophagy can be used in the elimination of *Mtb* [402]. LRG-47 initiates autophagy according to the study carried out by Singh *et al.* [366]. Immunity-related GTPase family M protein (IRGM) also plays role in autophagy and degradation of intracellular bacillary load.

Dendritic cells (DCs) play a vital role in the activation of the immune system on encountering a pathogen [330]. DCs are summoned to the lamina propria of the small intestine after bacterial infection. The number of DCs summoned depends on the pathogenicity of microorganisms confronted. Infection stimulates the release of a variety of soluble factors, including chemokines, which facilitate the summoning of DCs, and cytokines that are strong arbitrators of DC activation. Pathogens, viruses and their components can activate DCs directly. One of the important characteristics of DCs is their ability to migrate. During some infections, this property may have a harmful as well as a favorable side. Relocation of pathogen-laden DCs from the periphery into lymph nodes leads to the activation of T-cells. On the other hand, it contributes to the spread of infection within the host.

Glycoconjugates can alter the immune system of human body. Immunomodulatory components of *Mtb* are phosphatidyl-myo-inositol (PMI), lipomannan (LM) and lipoarabinomannan (LAM). Apart from LM and LAM, mannose also contributes to the synthesis of multiple glycosylated proteins and also polymethylated polysaccharides in *Mycobacteria* [281]. These molecules are synthesized by both pathogenic and non-pathogenic species. Many of the genes involved in biosynthesis of these glycoconjugates are important for survival of *Mycobacteria* [343, 344]. Only serine-threonine kinases have been predicted to take part in the regulation process of Mycobacterial glycosyltransferases [11, 283]. The interaction of *Mycobacteria* with the pattern recognition receptors may be an influencing factor for the functioning of the inflammatory signals, hence determining the way in which the immune system reacts [11, 283].

Iron plays a crucial role in the secretion of cytokines and in the activity of the transcription factors, affecting the immune response [116, 292]. Iron homeostasis is controlled by immune cell derived mediators and acute phase proteins. An effective method of host defense is to restrict the supply of iron to the pathogens. Pathogens have evolved to utilize iron as it is found plenty in the host. The control of iron homeostasis is one of the main issues, as it can be controlled by the host or the pathogen for their benefit.

With such kind of diverse mechanisms involved at each step of pathogen infection, predicting the host-pathogen interactions are extremely crucial. However, prediction of interactions among the huge number of host and pathogen proteins do pose a real-time experimental problem. Hence, many *in silico* prediction methods have been devised to abate such issues. They effectively provide the primary screening of the possible interactions and provide a list of highly probable interactions, which can then be experimentally verified. A few of them has been described and listed in the following section.

2.3 Methods for Prediction of Host-Pathogen Interactions

In this thesis, we concentrate on two crucial aspects to study host-pathogen interactions, viz., identification of toxin and analyzing their effect on host pathways. Multiple investigations have been done on the identification of toxins (effector proteins). However, not much can be said about the latter. The only research that has been done was related to PPIs. Multiple investigations report algorithms to predict binding of pathogen proteins to their host proteins. In this section, we describe various algorithms which facilitate the identification of toxins and analyzing the effect of toxins on pathways by predicting PPIs.

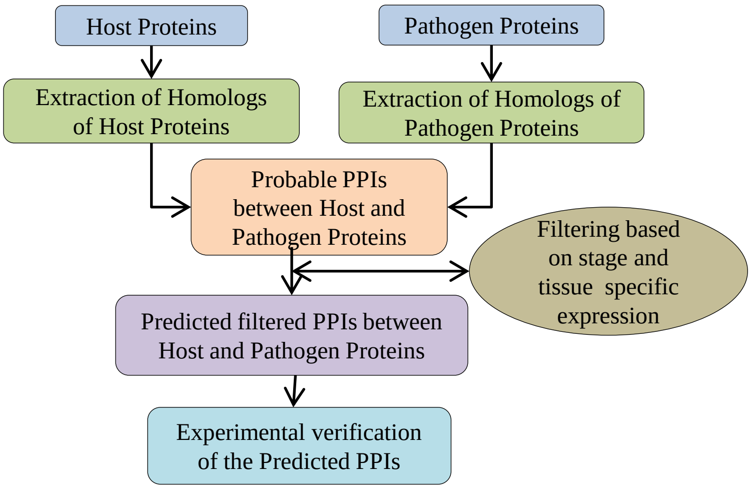
Prediction in the domain of host-pathogen interactions play a vital role in designing rational-therapeutic measures including drugs. Sometimes, experimental procedures can be cumbersome, time-consuming and expensive. Experimenting with all possibilities takes a lot of time. Prediction methods with the help of machine learning can overcome such problems. They can be used to predict the putative data first, which satisfies certain conditions. Then the predicted set can be verified experimentally, which will engage far less time and resources. The respective subsections describe some of the widely used techniques for *in silico* prediction of host-pathogen interactions. One or more of these methods can be used for prediction of genes, proteins, host factors and pathways, among others, of both the host and pathogen.

2.3.1 Biological reasoning-based prediction of host-pathogen interactions

The most extensively explored method by which a pathogen interacts with the host, is by PPIs. Pathogen proteins interact with host proteins for invading the host. Proteins of a pathogen can affect a host and its environment in multiple ways. They can directly bind with host protein(s) and affect downward cascades of reactions preventing normal function(s) of host. They can even compromise a host's immunological defenses by misguiding and weakening it. They can even utilize the components of a crumbling harsh anaerobic environment of a immune-compromised host. Hence predicting the putative PPIs between a pathogen and its host(s) is of paramount importance. In order to foretell whether a host protein can interact with a pathogen protein or vice-versa, the following categories of methods can be used.

Homology-based prediction An interaction between a pair of proteins in one species is anticipated to be conserved in its related species [268]. Prediction of host-pathogen PPIs in *Homo sapiens* (as host) and *Plasmodium falciparum* (as pathogen) [229] considers interaction templates of human and *P. falciparum* genomic sequences to bring out the probable set of PPIs. The homology detection algorithm as shown in Figure 2.3, is applied to these PPIs, to filter out non-homologous ones. The new set thus formed, is made to pass through the filter of stage specific and tissue specific expression data of *P. falciparum* and *Homo sapiens* respectively, and further filtered using the concept of predicted localized data. A study by Lee *et al.* [244] has considered orthologous pair of genes from 18 different species to predict

PPIs. Further analyzing them, 81 genes are found to be conserved in all the 18 species, 243 genes are missing in *P. falciparum* but found in the rest of 17 species. Hence, these 81 genes and their related PPIs are probably conserved.



Host Proteins
Pathogen Proteins
Extraction of Homologs of Host Proteins
Extraction of Homologs of Pathogen Proteins
Probable PPIs between Host and Pathogen Proteins
Filtering based on stage and tissue specific expression
Predicted filtered PPIs between Host and Pathogen Proteins
Experimental verification of the Predicted PPIs

Figure 2.3: Homology-based predictions of host-pathogen interactions

Homology-based approaches to host-pathogen PPI prediction are widely used for their sheer simplicity and biological background support. Since the data needed for implementing the predictions are only the template PPIs and protein sequences, these approaches are adaptable and can be applied to multiple different host-pathogen systems.

Similar is the case of molecular interaction between GBP (Galactose-Binding Protein) and LPS gram-negative bacterial Lipopolysaccharide). GBP from *Carcinoscorpius rotundicauda* performs as an anti-microbial defense [260]. Most importantly, GBP shares architectural and functional homology to human proteins. Therefore, there is a probability of some human protein and LPS interactions. Moreover, there are 6 Tectonic domains containing LPS binding sites in GBP. GBP acts as a bridge between LPS and CRP (C- Reactive Protein) by indulging in GBP-LPS and GBP-CRP interactions with the aim of forming a stable pathogen recognition molecule. These interactions have indicated that Tectonin domains can differentiate between host and pathogen proteins.

Homology-based approach have their own set of weaknesses. In an infection, two proteins in a predicted PPI may actually have very low probability to be present together. Therefore, host-pathogen PPIs predicted completely on the homology basis, without taking into consideration other biological properties of the proteins involved, may not be very dependable. Further information is needed to increase the accuracy of the prediction. An investigation by Wuchty and Stefan [426] has described filtering of the PPIs predicted by the homology-based approach using a random forest classifier. Then the result has been filtered according to expression and molecular characteristics. It has led to a potent subset of proteins that indeed interact.

Structure-based prediction When a pair of proteins have structures that are similar to a known interacting pair of proteins, it is justifiable to believe that the former are likely to interact in a way similar to the latter. Likewise, several investigations have used structural information to recognize the similarity between query proteins (i.e., proteins in the host and pathogen) and template PPIs (i.e., known interacting protein pairs), and conclude that host-pathogen protein pairs, which match some template PPIs, indeed interact. The method is depicted in Figure 2.4. A computational method for prediction of PPIs representing host-pathogen interactions has been devised by Davis *et al.* [99]. Their proposed method has first scanned the host and pathogen genome, searched for structural similarity to the already known protein complexes, and then analyzed their probable interactions, using the physical structures of the proteins. The result finally has undergone a filtering by tissue specific expression data of host proteins and stage specific expression data of pathogen proteins, leading to a potent set of proteins that have a high probability to interact.

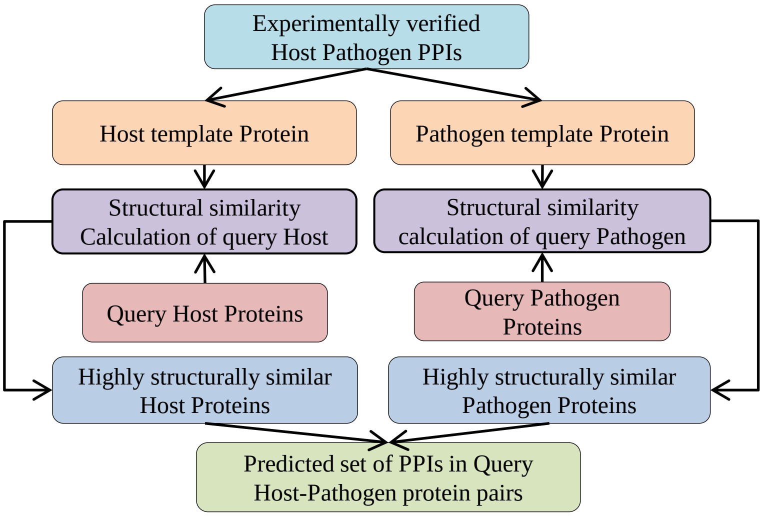

Experimentally verified Host Pathogen PPIs
Host template Protein
Pathogen template Protein
Structural similarity Calculation of query Host
Structural similarity calculation of query Pathogen
Query Host Proteins
Query Pathogen Proteins
Highly structurally similar Host Proteins
Highly structurally similar Pathogen Proteins
Predicted set of PPIs in Query Host-Pathogen protein pairs

Figure 2.4: Structure-based predictions of host-pathogen interactions

Mapping of PPIs between the dengue virus, and its human and insect host has been carried out by Doolittle *et al.* [118]. They have also predicted the interactions depending on structural similarity of the host and the pathogen proteins. It has also focused on predictions relevant to stress, unfolded protein response and interferon pathways. Another work by Dolittle *et al.* [117] has predicted PPIs between HIV-1 and *Homo sapiens* based on structural similarity. It has modeled a network of interactions between HIV-I and human proteins. Structurally similar proteins from host and HIV-1 has been retrieved, and from this structurally similar set of proteins, the known interactions have been mapped. The resultant subset has again been screened with factors, like cellular co-localization and RNAi screen to get a more determined set that has higher probability to interact. The result has highlighted a more potent set of proteins with higher chances of forming PPIs representing the interactions among human and HIV-1.

Domain/motif interaction-based prediction Here, the methodology for prediction of host-pathogen PPIs involves integration of known intra-species PPIs with protein domain profiles, and thereby predicting PPIs between a host and a pathogen [125]. For a set of intra-species PPIs, the functional domains are identified for each interacting proteins. For each pair of functional domain, Bayesian statistics is used to compute the possibility of two proteins to interact containing that pair of domain. The method is shown in Figure 2.5. It has been applied to *Homo sapiens-Plasmodium falciparum* host-pathogen system, and has successfully predicted 516 PPIs. Human proteins anticipated to interact with the same *Plasmodium* protein are close to each other in the human PPI network, and *Plasmodium* pairs predicted to interact with the same human protein are co-expressed in DNA micro-array datasets measured during various stages of the *Plasmodium* life cycle.

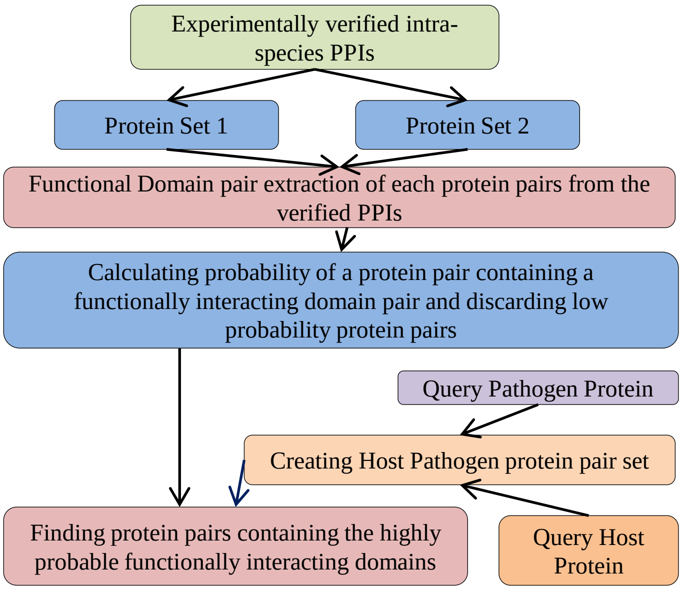

Experimentally verified intra-species PPIs
Protein Set 1
Protein Set 2
Functional Domain pair extraction of each protein pairs from the verified PPIs
Calculating probability of a protein pair containing a functionally interacting domain pair and discarding low probability protein pairs
Query Pathogen Protein
Creating Host Pathogen protein pair set
Finding protein pairs containing the highly probable functionally interacting domains
Query Host Protein

Figure 2.5: Domain/motif-based prediction of host-pathogen interactions

2.3.2 Machine learning-based predictions of host-pathogen interactions

Machine learning-based prediction methods are extensively used for detecting host-pathogen interactions as shown in Table 2.1. This table lists a few machine learning methods used for prediction of various aspects of host-pathogen interactions in different species. Moreover, the particular domain knowledge is also included in this table. The sub-area of research in some cases is referred as “Pathogen Informatics”. Supervised learning has been used for the prediction of PPIs in the host-pathogen domain by Tastan *et al.* [388]. The investigation has considered 35 features, including tissue distribution, gene expression profile, gene ontology, graph properties of human interactome, sequence similarity, post-translational modification similarity to neighbor and HIV-1 protein type features among others. Then the authors have selected the top 3 and top 6 features which are of maximum importance to classify the given data set into interacting and non-interacting classes. Random Forest classifier has been used as a tool for supervised learning with these feature set for training and resulting in MAP (Maximum a Posteriori) of 23%. From this computation, it has been concluded that graph and neighbor similarity features contribute to a better classification. Prediction of PPIs, based on motif conserved in HIV-1, has been performed by Evans *et al.* [136] and Bertoletti

Table 2.1: Summary of the machine learning-based tools used in the domain of host-pathogen interactions.

Machine Learning Method	Species	Reference	Domain
Random Forest Classifier	*HIV1-Homo sapiens*	Tastan *et al.* [388], 2009	PPI
Naive Bayes Classifier	Phylum *Chlamydiae* and genera *Escheria, Yersinia* and *Pseudomonas*	Arnold *et al.* [17], 2009	T3SS
Ensemble learning	*Legionella pneumophila*	Burstein *et al.* [59], 2009	T4SS
Support Vector Machine and Artificial Neural Network	*Pseudomonas syringae*	Löwer *et al.* [261], 2009	T3SS
Support Vector Machine	*Pseudomonas syringae*	Samudrala *et al.* [338], 2009	T3SS
Support Vector Machine	*Pseudomonas syringae*	Yang *et al.* [440], 2010	T3SS
Semi Supervised Learning using Multi-layer Perceptron	*HIV1-Homo sapiens*	Yanjun *et al.* [320], 2010	PPI
Support Vector Machine	*Ralstonia solanacearum*	Wang *et al.* [412], 2011	T3SS
Random Forest Classifier	*Homo sapiens-Plasmodium falciparum*	Wuchty [426], 2011	PPI
Group lasso with *l1/l2* regularization	*Homo sapiens-Salmonella, Homo sapiens-Yersinia*	Kshirsagar *et al.* [230], 2012	PPI
Support Vector Machine	None	Thieu *et al.* [392], 2012	Data Mining
Multi-task Classifier using Support Vector Machine	*Yersinia pestis, Francisella tularensis, Salmonella* and *Bacillus anthracis*	Kshirsagar *et al.* [231], 2013	PPI
Support Vector Machine	Multiple organisms	Zou *et al.* [456], 2013	T4SS
Ensemble learning	Multiple organisms	Wang *et al.* [408], 2017	T4SS
Ensemble learning	Multiple organisms	Wang *et al.* [406], 2018	T3SS
Deep learning	*Pseudomonas syringae*	Xue *et al.* [433], 2018	T3SS
Ensemble learning	Multiple organisms	Xiong *et al.* [432], 2018	T4SS

et al. [38]. The similarity between the binding motifs shared by virus and host proteins plays an important role in the crosstalk between virus and host. Similarly, the study by Bertoletti *et al.* [38] has attempted to predict PPIs based on motif conserved in HIV-1. It has also highlighted the role of chemokines as a factor for liver inflammation.

Computational prediction of T3 secreted effector proteins using machine learning techniques has been done previously [17,261,338,406,433,440]. Prediction of secretion signals in genomes of gram-negative bacteria has been done by Löwer *et al.* [261]. The authors have used SVM (Support Vector Machine) and ANN (Artificial Neural Network) with gradient descent back-propagation learning, momentum, and an adaptive learning rate to classify proteins as T3 effectors and non-effectors. Samudrala *et al.* [338] have predicted using SVM the mechanism of secreted substrates, and identified conserved secretion signal for T3 secretion systems. SVM has also been applied to N-terminal of amino acid sequences to predict novel T3 effector proteins [440]. Similarly, T3 secreted proteins have been predicted based on the amino acid sequences by Arnold *et al.* [17]. The authors have compared the performances of prediction made by naive Bayes classifier, 1-nearest neighbor, logistic regression, naive Bayes multinomial, SVM and voted perceptron methods. Wang *et al.* [406] have predicted T3 effector proteins, using a two-layered ensemble predictor Bastion3, based on the features obtained from N-terminal of the proteins. Xue *et al.* [433] have used deep learning framework to predict T3 effector proteins taking only the first 100 residues for prediction. In another attempt to predict bacterial Type III secreted (T3S) effectors, a distinct N-terminal position-specific amino acid composition feature has been found in more than 50% of T3S proteins [412]. Bi-profile Bayes method has been used in this particular work for feature extraction. Then the entire dataset along with the new feature has been analyzed with a new SVM-based classifier. The new classifier has classified T3SS and non-T3SS proteins successfully.

Identification of T4 effector proteins has been done on the basis of amino acid composition by Zou *et al.* [456]. The authors have used SVM to predict T4 effector proteins with an accuracy of 95.9%. The investigation has separately identified T4A and T4B effector proteins. Identification of T4 effector proteins in *Legionella pneumophila* has been done by using a machine learning approach [59]. The ORFs of the proteins in *Legionella pneumophila* have been classified as either effector or non-effector proteins. Genomic, evolutionary, regulatory networks and pathogenic attributes have been extracted from ORFs so as to identify T4 effector proteins. Xiong *et al.* [432] and Wang *et al.* [408] have predicted T4 effectors using ensemble classifiers based on only C-terminal features. The latter group has developed Bastion4 to predict T4 effectors. McDermott *et al.* [271] summarizes the computational prediction of T3 and T4 effector proteins, concluding that T3 secretion signals are similar across many different bacteria.

In order to establish a relation among a host and multiple pathogens, Kshirsagar *et al.* [231] have developed a method taking the similarity in infection initiated by four different pathogens in human host. The authors have used machine learning technique in the form of multi-task classification framework. The host-bacteria PPIs have been used as the input to the multi-task classifier, which has then classified the PPIs into interacting and non-interacting classes. Considering the biological hypothesis of similar pathogens targeting the same critical biological processes in a host, the classifier has minimized the empirical error on the training set and favored models that are biased towards the biological hypothesis. To prevent generation of a biased classifier, a bias term has been incorporated into the classifier in the form of regularizer.

A semi supervised multi-task method has been used on *Homo sapiens*-HIV 1 dataset [320] to predict host-pathogen PPIs. The method involves both supervised and semi-supervised learning. The supervised classifier has worked on labeled PPIs data. The semi-supervised classifier has shared network layers of the supervised classifier and got trained with partially labeled PPIs. This entire framework has been used to improve the recognition of interacting pairs. The supervised classifier has done multi-tasking with a semi-supervised classifier so that weak positive labels could ameliorate the supervised classification.

For prediction of PPIs between *Homo sapiens* and *Plasmodium falciparum*, a random forest classifier has assessed a set of PPIs, and then filtered the result according to expression and molecular characteristics, leading to a subset of proteins which indeed interact among themselves [426]. It has been observed here that the separate sets and a combined set of predicted and experimentally verified interactions have shared similar characteristics. In another investigation, Kshirsagar *et al.* [230] have tried to improve the supervised learning-based prediction of PPIs between *Salmonella*-human and *Yersinia*-human. This has been done by replacing the missing values of the dataset by the values generated by cross species information along with group lasso technique with regularization (obtained 77.6% precision). In order to impute values, localized-nearest neighbor approach (that uses sequence similarity) has been used as the basis to compute locality.

Data mining also forms an integral part of machine learning. Retrieved data about host-pathogen interactions in a few cases reflects information in two different ways, i.e., feature-based (SVM) [392] and language-based [76]. Chaussabel *et al.* [76] have used hierarchical clustering algorithm, by taking the literature available to identify functionally and transcriptionally homologous pair of genes as input. Removal of noise from the PPI databases has been done by removing PPIs that have less probability of taking place. Each such PPI has then been given a score. Then these PPIs have been hierarchically clustered to obtain the PPIs likeliness of occurrence. In this way, it has been found that out of 12122 binary PPIs obtained from BioGRID, 7504 PPIs are less likely to take place.

2.4 Online Repositories for Host-Pathogen Interactions

Host-pathogen interactions data can be obtained from several databases and repositories. These repositories have been summarized in Table 2.2. Some of these databases are referred purely for their data content, i.e., genome, proteome and metabolic pathway data [418], virus-virus, host-virus and host-host interaction networks [294], PPIs of hosts and pathogens [235], literature-based viral-human protein interactions [75], experimentally verified pathogenic, virulence and effector genes of fungal pathogens [423], human signaling and regulatory pathways [349], information on specific biodefense and public health pathogens [376], 3D viral proteins [359], information on invertebrate vectors of human pathogens [240], and a collection of genus specific databases [24] among others. Some of these databases even have integrated in-house tools, i.e., BLAST interface [119] and browser [454] for host-pathogen interactions data analysis. Some tools [137] used in analysis and visualization of these kinds of data, has been described below.

PAThosystems Resource Integration Center (PATRIC) [418] includes a relational database, analytical pipelines, and a website that supports querying, browsing, data visualization, and allowing the download of raw and curated data in standard formats. Currently, the database houses complete sequences for viral and bacterial genomes, hence providing an all-inclusive bioinformatics resource for pathogens.

Pathway Interaction Gateway (PIG) provides a text-based search and a BLAST interface for searching the host-pathogen PPIs. Each entry in PIG incorporates information on the functional annotations and the domains present in the interacting proteins [119].

VirHostNet (Virus-Host Network) [165, 294] is a public knowledge base specialized in the management and analysis of integrated virus-virus, host-host and virus-host interaction networks coupled with their functional annotations. VirHostNet contains data of virus-host and virus-virus interactions constituting more than 180 distinct viral species. VirHostNet Web interface provides suitable tools which allow effective query and visualization of infected cellular network.

HPIDB (Host-Pathogen Interaction Database) [235] primarily contains experimentally verified and predicted PPIs of hosts and pathogens.

GPS-Prot [137] is a software tool that permits users to easily create an all-inclusive and integrated HIV-host networks. Its web-based format, which requires no software installation or data downloads, gives it an extra edge over other visualization tools. GPS-Prot enables users to quickly generate networks that amalgamate both genetic and protein-protein interactions between HIV and its human host, into a single representation.

VirusMint [75] contains protein interactions between viral (papilloma viruses, HIV-1, Epstein-Barr, hepatitis B, hepatitis C, herpes and Simian virus 40) and human proteins reported in the literature. VirusMINT presently stores interactions constituting more than 490 unique viral proteins from more than 110 different viral strains.

PHIDIAS (a Pathogen Host Interaction Data Integration and Analysis System) [429] is a database and analysis system to curate, analyze and address different scientific issues in the areas of host-pathogen interactions (PHI, or called host-pathogen interactions or HPI).

MvirDB [454] integrates DNA and protein sequence information from multiple databases. Entries in MvirDB are hyper-linked back to their original sources. A blast tool enables the user to blast against all DNA or protein sequences in MvirDB, and a browser tool enables the user to explore the database to retrieve virulence factor descriptions, sequences and classifications, and to download sequences of interest. PHI-base [423], a web-accessible

Table 2.2: List of online repositories storing data related to host-pathogen interactions

No.	Name	URL
1	PATRIC [418]	http://patricbrc.org/portal/portal/patric/Home
2	PIG [119]	http://patricbrc.org/portal/portal/patric/HPITool
3	VirHostNet [294]	http://virhostnet.prabi.fr/
5	HPIDB [235]	http://agbase.msstate.edu/hpi/main.html
6	GPS-Prot [137]	http://gpsprot.org/
7	VirusMint [75]	http://mint.bio.uniroma2.it/virusmint/Welcome.do
8	PHIDIAS [429]	http://www.phidias.us/introduction.php
9	MvirDB [454]	http://mvirdb.llnl.gov/
10	PHI-base [423, 424]	http://www.phi-base.org/
11	PID [349]	http://pid.nci.nih.gov/
12	BioHealthBase [376]	http://www.biohealthbase.org/
13	VPDB [359]	http://www.vpdb.bicpu.edu.in/
14	VectorBase [240]	https://www.vectorbase.org/
15	EuPathDB [24]	http://eupathdb.org/eupathdb/
16	PHISTO [390]	http://www.phisto.org/
17	ViPR [316]	http://www.viprbrc.org/brc/home.spg?decorator=vipr
18	EDWIP [303]	http://cricket.inhs.uiuc.edu/edwipweb/edwipabout.htm
19	HoPaCI-db [42]	http://mips.helmholtz-muenchen.de/HoPaCI
20	VFDB [80]	http://www.mgc.ac.cn/VFs/main.htm
21	AquaPathogen X [133]	http://pubs.usgs.gov/fs/2012/3015/
22	MorCVD [365]	http://morcvd.sblab-nsit.net/
23	Mtb-HID [309]	http://www.pantlab.co.in/mtb-hid/
24	PHISTO [124]	http://www.phisto.org/
25	KEGG [209]	https://www.kegg.jp/
26	PDB [36]	https://www.rcsb.org/

database currently catalogs experimentally verified virulence and effector genes from fungal and oomycete pathogens. These pathogens interact with animal, plant and fungi as hosts.

PID [349] is a freely available collection of curated and peer-reviewed pathways composed of human molecular signaling and regulatory events and key cellular processes. PID offers a range of search features to facilitate pathway exploration.

BioHealthBase [376] is a public bioinformatics database and analysis resource for study of specific biodefense and public health pathogens, like *Francisella tularensis*, *Mycobacterium tuberculosis*, *Influenza virus*, *Microsporidia* species and ricin toxin. It serves as a substantial integrated repository of data imported from public databases and data derived from various computational algorithms and information curated from the scientific literature. Its 3D visualization capacity allows researchers to view proteins with their key structural and functional features highlighted.

VPDB (Viral Protein Structural Database) [359] is an interactive database for three-dimensional viral proteins. It provides an all-inclusive resource, with an emphasis on the description of derived data from structural biology. At present, VPDB includes viral protein structures from more than 277 viruses with more than 465 virus strains.

VectorBase [240, 241, 276] is a web-accessible data repository storing information about invertebrate vectors of human pathogens. It annotates and maintains vector genomes providing an integrated resource for the research community. It hosts data related to 9 genomes, i.e., mosquitoes (3 *Anopheles gambiae* genome), *Aedes aegypti* and *Culex quinquefasciatus*), body louse (*Pediculus humanus*), tick (*Ixodes scapularis*), tsetse fly (*Glossina morsitans*) and kissing bug (*Rhodnius prolixus*). The data spans across genomic features, expression data, population genetics and ontologies.

EuPathDB [23, 24] is an integrated database covering the eukaryotic pathogens of the genera *Giardia, Cryptosporidium, Neospora, Leishmania, Toxoplasma, Plasmodium, Trypanosoma* and *Trichomonas*. These groups are supported by a taxon-specific database built upon the same infrastructure. EuPathDB portal provides an entry point to all these resources, and the opportunity to leverage orthology for searches across genera.

Similarly, a number of other databases, like PHISTO [390], ViPR [316], HoPaCI-DB [42], VFDB [80] [436] [79], EDWIP [303], AquaPathogen X [133], MorCVD [365], Mtb-HID [309], PHISTO [124], are available, which help in the host-pathogen interactions domain research.

2.5 Discussion and Future Scope

This section has discussed multiple faucets of host-pathogen interaction research, the shortcoming of the previously defined methodologies as discussed in Sections 2.2 and 2.3 and

Table 2.3: Summary of host protection and pathogen attacking mechanisms.

Host Protection Mechanism	Pathogen Attacking Mechanism
Protein-Protein Interactions (GBP galactose-binding protein)	Protein - Protein Interactions (target hub protein)
shRNAs (pathogen gene knock down)	microRNAs (protection against cellular micro-viral response,gene silencing)
Autophagy	MAM (multivariate adhesion molecule, high binding affinity with host during infection)
siRNAs (inhibit HIV-1 replication)	Pneumolysin (virulence factor)
Macrophages	Inhibition of macrophage
Restricting supply of Iron	Glial cells of host (convert it into progenitor cells then survive in the host)
None	Motif mimicry (utilized by pathogens to rewire host pathways by co-opting SLiM mediated protein interactions)
None	Biofilm formation
None	Hijacking (implant own sequence in normal sequence of host)

the future scopes associated with the aforesaid methodologies. It takes both the host and pathogen points of view into account. The ways in which a pathogen can attack its host, the proteins emitted by a pathogen responsible for perturbing normal functionality of host, the genes responsible for such proteins, silencing and hijacking gene mechanism of pathogens, inhibiting the functions of macrophages, along with genes and proteins needed for their survival inside a host has been discussed. From the hosts point of view, the factors of pathogen that activates immune response has also been discussed. Salient features of the discussion is given in Table 2.3. The genes of multiple strains of an organism have been studied in several investigations [103, 270, 301] to understand the infection mechanism of these strains on the host, and to locate the difference between them. In order to survive in a host, a pathogen can either perform hijacking [328] or it can use the existing environment to survive [50]. The effect of the genes in different strains of a pathogen has been studied. There is still uncertainty in the generalization/specialization of interactions in different strains of pathogens. A study has suggested that different strains of the same pathogen have different methods of invasion [270]. On the contrary, a counter example has also been provided in [91], which indicates that two strains of *Mycobacterium* have homologous genes required for Snm.

Influenza, DENV-2 and HIV have been in the limelight for identification of the host factors. Other pathogens too need to be taken into account. Inhibition of macrophage is a prospective aspect of research in bioinformatics. The inhibition mechanism needs to be studied in more pathogens apart from the mostly studied ones to find similarity between the inhibition mechanisms among these organisms.

Machine learning-based prediction methods have been applied mainly to PPIs. However, protein-ligand interactions and hence prediction of pathways via machine learning methods have not been critically investigated. Different pathogens become drug resistant and form new pathways, and these newly formed pathways can perturb the present host pathways in an unknown way. Similarly, machine learning algorithms in the field of pathway predictions are needed, which would mainly consider protein-ligand binding. Additionally, reaction dynamics are needed to be thoroughly examined, as pathways are nothing but chain of reactions. Prediction of Type III secreted bacterial proteins by machine learning techniques is also a challenging task. However, a major drawback in the area of prediction of host-pathogen PPIs, are the unavailability of data sets for different pathogens. Moreover, there is always this lurking issue of biological validation of the predicted PPIs.

Some of the organisms studied for the exploration of host-pathogen PPIs are *Homo sapiens-Plasmodium falciparaum* [125, 229, 244, 426], *Homo sapiens*-Dengue virus [118], *Homo sapiens*-HIV 1 [38, 117, 136]. However, there are many more host-pathogen pairs waiting in the line for these kinds of studies. In addition, homology-based approaches have their own inherent weaknesses. In real scenario, two proteins in a predicted PPI may actually have little opportunity to be present close enough to interact with each other. Therefore, host-pathogen PPIs predicted entirely on the basis of homology, without considering other biological characteristics of the proteins involved, may not be reliable. Additional information must be used to increase the accuracy of the prediction and make the predictions biologically sound. Keeping this in mind, the study by Wuchty [426] has filtered the predicted PPIs based on homology using gene expression and molecular characteristics. It has led to the formation of a concrete set of PPIs closely mimicking the biological scenario. The prediction of PPIs by comparative modeling [99], have very stringent filters leading to the formation of a smaller and robust set of PPIs.

Supervised, unsupervised and semi-supervised learning have been mostly used for prediction of host-pathogen PPIs. The organisms for which these predictions have been made are mainly *Homo sapiens-HIV1* [320,388], *Homo sapiens-Plasmodium falciparum* [426] and *Homo sapiens-Saccharomyces cerevisiae* [88]. Both Tastan *et al.* and Yanjun *et al.* [320,388] have applied their respective algorithms on the same dataset which restricts the contribution of the articles. The performance of random forest-based classifier is negligibly better than the multilayer perceptron [320]. Some researches have selected the top 6 and top 3 features among 35 features to predict whether a protein is interacting or not [388]. However, doing so may not give an accurate prediction since the interaction between proteins depends on all of its features even if by negligible amount which should not be ignored.

A flaw is often noticed in the choice of datasets. In a semi-supervised learning approach to identify PPIs [320], the negative dataset is comparatively more extensive than the positive one. The negative (non-interacting) data set has approximately 16000 pairs of proteins while the experimentally verified positive (interacting) dataset has only 158 pairs of protein. Training with such a dataset might lead to a biased classifier and the classifier would be inclined to predict most test pairs as non-interacting. Moreover, the logic used behind selecting non-interacting dataset is based on a random list of pairs of proteins which do not fall into the positive set. It is always a risk, since there is no experimental evidence that the selected negative pairs will not interact. There may be several interacting pairs present among the negative set. Another study has been done for predicting proteins secreted by Type III secretion system based only on structural and compositional aspect of the proteins [17]. These studies should include other factors, like expression and molecular characteristics.

One notable aspect of *in silico* analysis of host pathogen interactions is that hardly any research has been carried out to study the effect of perturbation on metabolic and signaling pathways. If enzyme(s) from a pathogen is introduced into a host, there is a possibility of these enzymes to get involved with more than one host pathways. There are no mechanisms available which would take a list of protein (enzyme) names and provide the pathway (just one pathway based on these enzymes) based only on those enzymes (at least 90%). Moreover, a pathogen can be associated with more than one disease. Such diseases, for which a pathogen is responsible, need to be looked into. The scenario becomes more complex, when a host suffers from two or more diseases (comorbidity) simultaneously, it implies presence of multiple pathogens responsible for multiple diseases in a host in real time. This kind of real-time simulation studies have hardly been done. However, analysis of such complex systems are of immense importance. For example, the disease COVID-19, caused by the virus SARS-CoV-2, in patients with any comorbidity has yielded poorer clinical outcomes than those without comorbidity [162].

An important aspect that needs to be considered is that some pathogenic proteins prevent the working of macrophage. This is a serious problem in host-pathogen domain. Drugs are needed that would facilitate the working behavior of a macrophage. Drugs are also needed for the prevention of formation of intracytoplasmic vesicle that HIV-1 uses to prevent identification by macrophages [82]. Formation of biofilm [288, 382] is another aspect that needs to be investigated. Breaking the biofilm formed by pathogens is indeed recommended to avoid the spread of infection. More attention is needed in this domain, given the rate at which new infectious pathogens are emerging along with their variety of degree of infection.

Hardly any research have been done based on the automated image processing-based techniques available for predicting host-pathogen interactions. A study by Mech *et al.* [273] has come up with a technique of a more robust analysis of microscopy images of macrophages that are made to coexist with different *A. fumigatus* strain. Usually the images are manually analyzed, which is time consuming and error prone. The authors used the feature set which includes size, shape, number of cells and cell-cell contacts. By analyzing the images, it has been found that different mutants of *A. fumigatus* have an impact on the ability of the macrophages to adhere and phagocytose the conidia. It has been observed that the rate of phagocytosis is higher in pksP mutant of *A. fumigates*, while it is not the same case in the other strains.

2.6 Conclusions

This chapter has covered various aspects of host-pathogen interactions. Interaction of a pathogen with its host(s) is always a unique mechanism. Each one of the pathogenic species has specific mechanism(s) to interact with their host. The different mechanisms of a number of species have been included in this survey along with the similarities in the attacking mechanism(s) of pathogens. The survey has introduced a brief history and introduction of the host-pathogen interactions research field followed by classification of host-pathogen interactions based on gene(s), protein(s), host-factor(s), involved pathway(s) and inhibition mechanism of macrophage(s). It has listed prediction methods used in the host-pathogen interactions domain based on biological reasoning (homology, structure and motif interaction), machine learning (unsupervised, semi-supervised and supervised) and sometimes both the methods. Various data sources used for research in this domain have also been listed. The survey concludes with a general discussion of the topic and future scopes followed by a conclusion. The field of host/pathogen interactions is emerging as a crucial area of infectious disease research in the post-genomic era. It is a budding research field where new discoveries are getting announced almost each day throughout the globe. The discovery of dynamics of the host-pathogen interactions will aptly facilitate further development in the field of discovering new drugs and new therapies for different diseases.

While conducting the survey, it has come to our attention that in the field of *in silico* analysis of host-pathogen interactions, the research is concentrated mainly on predicting new PPI's or identification of T3 and T4 effector proteins. No investigations have been reported to identify effector proteins excluding T3 and T4. Analyzing the effect of such toxins on signaling and metabolic pathways have not been looked into. In this regard, the thesis is dedicated to the identification of toxins, and analyzing their effect on signaling and metabolic pathways of the host by developing novel algorithms.

The following chapters are dedicated to the development of algorithms for identification of toxins released by the pathogens and their effect on host pathways based on feature extraction, classification and pathway prediction. The thesis can primarily be divided into two parts. The first part deals with the identification of the toxic perturbing agents. The second part deals with the analysis and identification of perturbations caused by these perturbing agents. Identification of toxins is the primal step towards the identification of disruption of biological pathways by such toxic perturbing agents from pathogens. As mentioned in Section 2.3.2, most of the investigations related to toxin identification has been focused on T3SS and T4SS effector proteins. The next chapter develops a new system, called PyPredT6, which identifies T6 effector proteins using a novel feature set. However, while experimenting, it has been noticed that the prediction tools for T3, T4 and T6 effector proteins have used primary and secondary structures for effector predictions.

In Chapter 4, a methodology for prediction of individual classes of T3, T4 and T6 effector proteins, a composite class of T1, T2, T5, T7 effector proteins, and a class of non-effector proteins based on their tertiary structures has been developed. While attempting to train a classifier, it has been noticed that the dataset is heavily imbalanced, and the state-of-the-art oversampling algorithms were not fit for our dataset. Consequently, we further developed a new oversampling algorithm which facilitates the accurate prediction of effector proteins.

While working on tertiary structure-based prediction, it has been noticed that another type of effector proteins, secreted by gram-positive pathogenic bacteria, has not been well researched. Hence in Chapter 5, we have developed a deep neural network-based system to predict T7 effector proteins in gram-positive bacteria using a unique feature set. We have not been able to consider the tertiary structures since not enough tertiary structures of T7 effectors were available in the literature. Hence, we had to go about with the primary and secondary structure information.

After the identification of toxins, we move on to identify and analyze their effect on biological networks. In Chapter 6, we focus on the effect of toxins on metabolic pathways. It gives an insight into how the presence of toxins affect the metabolic network. In order to achieve this goal, we have developed an algorithm that predicts pathways based on the structural similarity of metabolites. Using the novel algorithm, one can predict unknown transformations among a set of metabolites. The algorithm can be used to predict the changes in a metabolic pathway (in terms of metabolites being knocked out from the pathways) when it is exposed to such toxins.

The next task, after analyzing the effect of toxins on metabolic pathways, we have considered, is to analyze the effect of toxins on signaling pathways, by developing an algorithm in Chapter 7. The algorithm developed finds the stability of a network, the ability of a network to withstand perturbations, and the effect of toxins on the stability of the network. We have come up with a parameter to measure the robustness of biological networks before and after being perturbed by toxins released by pathogens. The algorithm uses this parameter to predict the robustness of a network.

Chapter 3

PyPredT6: An Ensemble Learning-based System for Identification of Type VI Effector Proteins [353]

3.1 Introduction

From the survey conducted in the previous chapter, it came to our attention that the effector proteins secreted by Type VI secretion system need an accurate identification system. Therefore, in this chapter we have developed such a system that would predict effector proteins of Type VI secretion system. Gram-negative bacteria have six different secretion systems, *viz*, Type I (T1SS), Type II (T2SS), Type III (T3SS), Type IV (T4SS), Type V (T5SS) and Type VI (T6SS) secretion systems [93]. These systems facilitate the transfer of certain proteins, known as "effector proteins", a type of toxin, required for bacterial growth and infection in the host environment.

The effector proteins, a type of toxin, play an important role in bacterial pathogenesis [339, 395], due to which their *in silico* identification is crucial. Effector proteins of gram-negative bacteria are translocated into host cells predominantly by T3SS, T4SS and T6SS [337] [446] [32] [439]. Among these secretion systems, Type VI (T6SS) secretion system has been discovered in the year 2006. T6SS associated effector proteins in many gram-negative bacteria are yet to be discovered. T6SS has also been found in pathogenic species, *viz.*, *V. cholerae*, *E. tarda*, *P. aeruginosa*, *B. mallei* and *F. tularensis* among others. T6SS has been identified to play a major role in the pathogenesis of *A. hydrophila* [381]. T6SS locus (YPO0499-YPO0516) has been found to play a crucial role in phagocytosis-promoting activity [332]. T6SS has also been discovered in plant pathogens, *viz.*, *A. tumefaciens*, *P. atrosepticum* and *X. oryzae* among others. Genes encoding T6SS have also been found in some non-symbionts such as *M. xanthus*, *D. aromatica* and *R. baltica*, where they may contribute to biofilm formation. *In silico* prediction of these proteins will facilitate faster experimental validation and provide clear information regarding pathogen invasion mechanisms via T6SS.

T6SS is a phage-tail-spike-like injectisome. The injectisome releases effector proteins directly into the cytoplasm of host cells [69]. Genes for T6SS components have been found in proteobacteria, planctomycetes, and acidobacteria. A few attempts have already been made towards *in silico* prediction of effector proteins of T3SS and T4SS [354]. Computational prediction of T3 secreted effector proteins using machine learning techniques has been done previously [17, 261, 338, 406, 433, 440]. Prediction of secretion signals in genomes of gram-negative bacteria has been done by Löwer *et al.* [261]. The authors have used SVM (Support Vector Machine) and an ANN (Artificial Neural Network) with gradient descent back-propagation learning, momentum and an adaptive learning rate to classify proteins as T3 effector proteins and non-effector proteins. Samudrala *et al.* [338] have predicted using SVM the mechanism of secreted substrates, and identified conserved secretion signal for T3 secretion systems. SVM has also been applied to N terminal of amino acid sequences to predict novel T3 effector proteins [440]. Similarly, T3 secreted proteins have been predicted based on the amino acid sequences by Arnold *et al.* [17]. The authors have compared the performances of prediction made by naive Bayes classifier, 1-nearest neighbor, logistic regression, naive Bayes multinomial, SVM and voted perceptron methods. Wang *et al.* [406] have predicted T3 effector proteins, using a two-layered ensemble predictor Bastion3, based on the features obtained from N-terminal of the proteins. Xue *et al.* [433] have used deep learning framework to predict T3 effector proteins taking only the first 100 residues for prediction.

Identification of T4 effector proteins has been done on the basis of amino acid composition by Zou *et al.* [456]. The authors have used SVM to predict T4 effector proteins with an $accuracy$ of 95.9%. The investigation has separately identified T4A and T4B effector proteins. Identification of T4 effector proteins in *Legionella pneumophila* has been done by using a machine learning approach [59]. The ORFs of the proteins in *Legionella pneumophila* have been classified as either effector or non-effector proteins. Genomic, evolutionary, regulatory networks and pathogenic attributes have been extracted from ORFs so as to identify T4 effector proteins. Xiong *et al.* [432] and Wang *et al.* [408] have predicted T4 effectors using ensemble classifiers based on only C-terminal features. The latter group has developed Bastion4 to predict T4 effectors. McDermott *et al.* [271] summarizes the computational prediction of T3 and T4 effector proteins, concluding that T3 secretion signals are similar across many different bacteria.

Bastion6, an SVM-based protein predictor, is currently the only available tool for prediction of T6 effector proteins [409]. However, multiple limitations have been noticed in the implementation of Bastion6 in terms of its dataset size, choice of non-effector proteins, choice of the classifier, speed of execution, reliability of the results, functionality of the server, its predicted effectors among others. Apart from Bastion6, Zalguizuri *et al.* [445] and An *et al.* [13] have made an attempt to predict T6 effectors. However, the results were unsatisfactory and they expressed a dire need for specialized models for T6 effector prediction. Moreover, these two investigations have considered the non-effector sets for T3 and T4 together as the non-effector set for T6. As mentioned before, due to multi-functional nature of proteins, T3/T4 non-effectors need not necessarily be T6 non-effectors.

Some notable demerits in the prediction of effector proteins (T3, T4, T6) by various investigations are that hypothetical proteins have been used in training data. In some of the investigations, non-effector dataset has been derived by choosing secreted proteins obtained from any of T1SS through T8SS in gram-negative bacteria. For creating the non-effector set in T3/T4 prediction, secreted proteins of types T1SS through T8SS, except T3/T4, have been taken into consideration. It may result in a non-effector list containing effector proteins since proteins are multi-functional in nature. Moreover, none of the aforesaid investigations have applied feature selection on their datasets, which elevates the risk of over-fitting.

In order to overcome the shortcomings of the above methods including Bastion6, we have developed PyPredT6, a standalone system for predicting probable T6 effector proteins using a set of unique 837 features derived from existing biological knowledge-base, *viz.*, SecReT6 [253] and SecretEPDB [14]. The chapter is organized as follows. We first describe the process of feature extraction to form the feature set. This is followed by a discussion of various preprocessing methods which the dataset has been subjected to. We further go on to describe the architecture of PyPredT6. In order to analyze the efficacy of PyPredT6, an elaborate description of biological validation of predictions of T6 effectors in *Vibrio cholerae* and *Yersinia pestis* by PyPredT6 has been provided. A detailed comparison of PyPredT6 with Bastion6 has been furnished subsequently. A discussion on the applicability and future scopes of PyPredT6 concludes the chapter.

3.2 Methodology

In this section, we elaborate the development of PyPredT6. PyPredT6 is a standalone system that reads nucleotide and amino acid sequences of unknown proteins in FASTA format (Appendix B.1) to identify whether these proteins are T6 effectors or not. The execution of PyPredT6 starts with the data collection phase where nucleotide and peptide sequences of effector and non-effector proteins are accumulated. Following the data collection phase, hypothetical and putative proteins are filtered out from the sequences.

The amino acid and nucleotide sequences corresponding to a protein contain intrinsic information that dictates its properties. These include composition of amino acids, order of amino acids, secondary structure-based information, solvent accessibility-based information and physicochemical properties. While each of the properties may contribute to the characteristics of T6 effectors, none of the properties alone can be a sufficient and necessary determinant for a protein to be a T6 effector. Thus, extracting features from these properties would better characterize T6 effectors. From the filtered effector and non-effector amino acid and nucleotide sequences, a spectrum of features has been extracted. The extracted features have been integrated to form a feature set using which PyPredT6 has been designed. In order to rectify the issue of data imbalance, Borderline-SMOTE is employed on the unbalanced training set. To prevent overfitting and improve the generalization performance of PyPredT6, feature selection has been implemented based on Gini impurity index.

Following feature extraction and selection, appropriate classifiers have been chosen based on which PyPredT6 has been developed. An ensemble learning system based on the consensus of Multi-Layer Perceptron (MLP), Support Vector Machine (SVM), k-Nearest Neighbor (kNN), Naive Bayes (NB) and Random Forest (RF) classifiers, via majority voting [7], has been employed for the identification of T6 effector proteins. Cross-validation has been implemented for parameter tuning of the individual classifiers in the ensemble. PyPredT6 is rigorously trained and tested for accurate identification of T6 effector proteins. The overall methodology followed in designing PyPredT6 is presented as a flowchart in Figure 3.1.

3.2.1 Data collection

We have accumulated a set of experimentally verified amino acid sequences of T6 effector proteins in different organisms from two databases, *viz.*, SecReT6[1] [253] and SecretEPDB[2] [14]. The corresponding nucleotide sequences have also been considered. A total of 175 unique effector proteins has been obtained from the databases. The non-effector set has been constructed from the entire genome of non-pathogenic gram-negative bacteria *Bacteriodes vulgatus* (*B. vulgatus*) [173]. An argument may arise here that housekeeping proteins of the same pathogenic species (from which effector proteins have been taken) provide a better option for the non-effector proteins. However, in prokaryotes, genes are often found to be multi-functional in nature [206, 314]. In order to avoid a housekeeping gene of the same species, which has some kind of direct or indirect association with an effector gene [341], here we have considered a different non-pathogenic gram-negative bacteria (*B. vulgatus*) which lives in human gut. It has to be mentioned here that the T6 effector protein set considered here comprises proteins from multiple gram-negative species.

A set of 4183 genes and their corresponding proteins of *B. vulgatus* has been obtained

[1] http://db-mml.sjtu.edu.cn/SecReT6/

[2] http://secretepdb.erc.monash.edu/

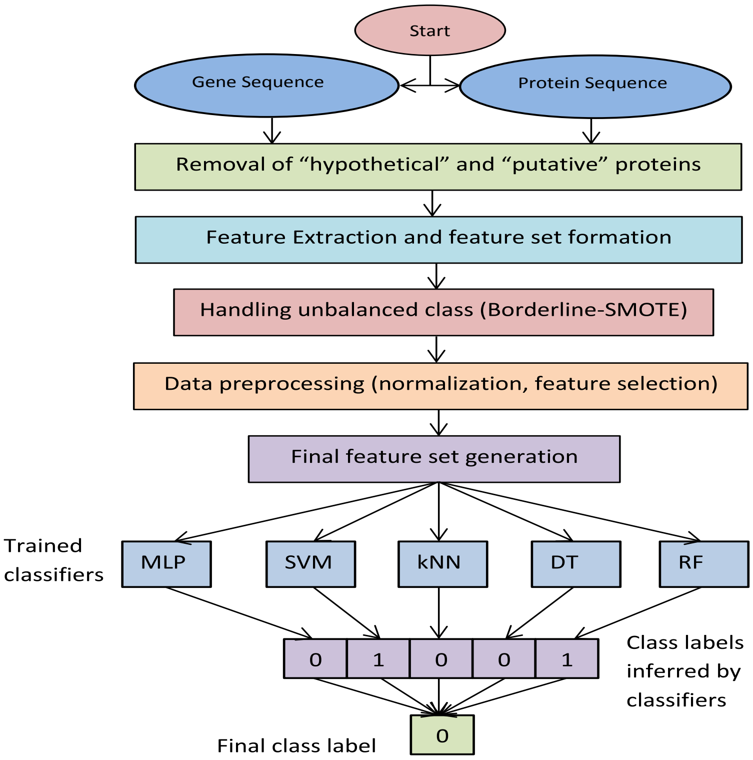

Start
Gene Sequence
Protein Sequence
Removal of "hypothetical" and "putative" proteins
Feature Extraction and feature set formation
Handling unbalanced class (Borderline-SMOTE)
Data preprocessing (normalization, feature selection)
Final feature set generation
Trained classifiers
MLP
SVM
kNN
DT
RF
0
1
0
0
1
Class labels inferred by classifiers
Final class label
0

Figure 3.1: Methodology for prediction of putative T6 effector proteins. A value of 1 indicates that a protein is pathogenic while 0 stands for a protein being non-pathogenic. Here an example of final class label of 0 is provided, based on majority voting of outcomes of the classifiers

from KEGG[3]. Proteins annotated as "putative","hypothetical" and "uncharacterized" have been removed from the set as no physical, genetic or functional annotation is available for such proteins. Thus a total of 1063 putative, 1572 hypothetical and 51 uncharacterized proteins have been removed from the initial set. Finally, we have considered 1497 non-effector proteins of *B. vulgatus*.

3.2.2 Feature extraction

In this section, we derive nucleotide and amino acid-based features from the sequences of T6 effector and non-effector proteins. A schematic representation of the feature set has been given in Figure 3.2.

Position specific nucleotide sequence profile (PSNSP): These features have been extracted from the nucleotide sequences of the genes. The percentage composition of 4 mononucleotides (A, T, G, C) in a gene, i.e., the percentage of each of A, T, G and C with respect to the total number of nucleotides in the sequence of a gene form position-specific mononucleotide sequence profile (PSMNSP). Likewise, the percentage composition of 16

[3] https://www.genome.jp/dbget-bin/get_linkdb?-t+genes+gn:T00546

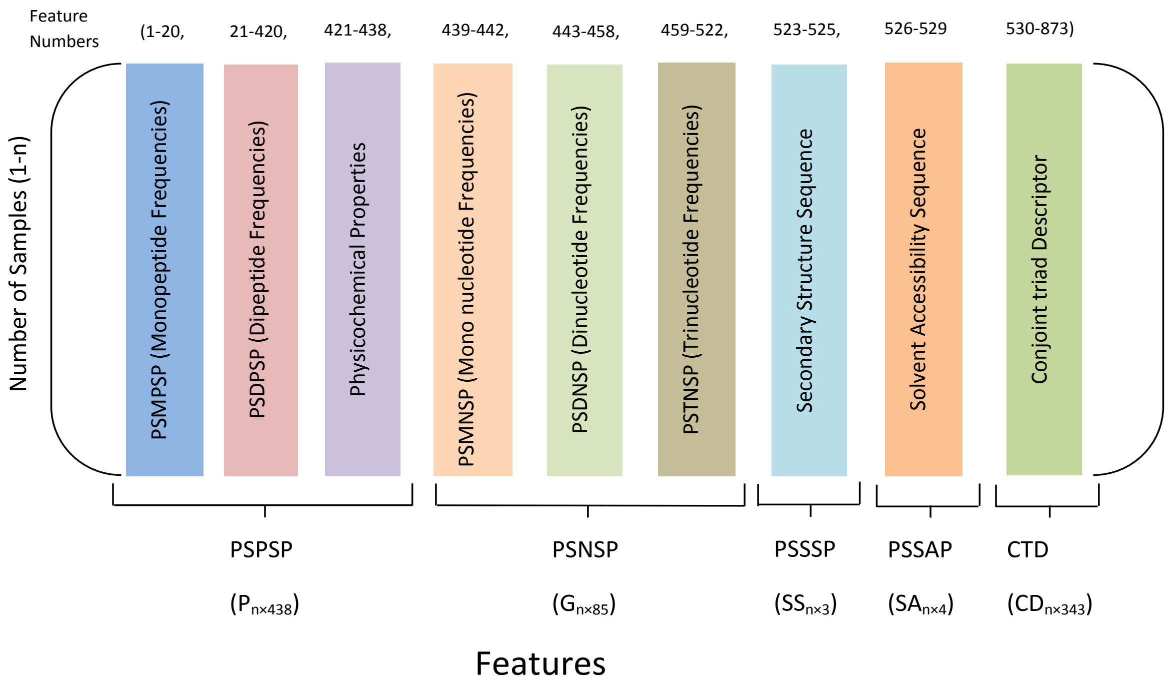

Feature Numbers
(1-20,
21-420,
421-438,
439-442,
443-458,
459-522,
523-525,
526-529
530-873)
Number of Samples (1-n)
PSMPSP (Monopeptide Frequencies)
PSDPSP (Dipeptide Frequencies)
Physicochemical Properties
PSMNSP (Mono nucleotide Frequencies)
PSDNSP (Dinucleotide Frequencies)
PSTNSP (Trinucleotide Frequencies)
Secondary Structure Sequence
Solvent Accessibility Sequence
Conjoint triad Descriptor
PSPSP
(P_{n×438})
PSNSP
(G_{n×85})
PSSSP
(SS_{n×3})
PSSAP
(SA_{n×4})
CTD
(CD_{n×343})
Features

Figure 3.2: The structure of the feature matrix where $G_{n\times85}$: gene feature matrix, $P_{n\times438}$: protein feature matrix, $CTD_{n\times343}$: conjoint triad descriptor matrix, $SS_{n\times3}$: secondary structure feature matrix, and $SA_{n\times4}$: solvent accessibility feature matrix.

di-nucleotides (AA, AT, AG, ..., and others) with respect to the total number of dinucleotides in the gene sequence form the position-specific dinucleotide sequence profile (PSDNSP). The percentage composition of 64 tri-nucleotides (AAA, AAT, AAG, ..., and others) with respect to the total number of triplets form position-specific trinucleotide sequence profiles (PSTNSP). Thus position specific nucleotide sequence profile (PSNSP) of a gene comprises PSMNSP (4 features), PSDNSP (16 features), PSTNSP (64 features), and GC content. In this way, we have got 85 features for a gene. These features altogether constitute the gene feature matrix $G_{n\times85}$ for n gene sequences.

Position specific peptide sequence profile (PSPSP): These features have been extracted from the protein sequences. The percentage composition of 20 single amino acids (A, G, H, ..., and others) in a protein, i.e., the percentage of each of A, G, E, V, I, L, F, P, Y, M, T, S, H, N, Q, W, R, K, D and C with respect to the total number of peptides in the sequence of the protein form the position-specific monopeptide sequence profile (PSMPSP). Likewise, the percentage composition of 400 di-peptides (AA, AG, AH, ..., and others) with respect to the total number of dipeptides in the protein sequence form the position-specific dipeptide sequence profile (PSDPSP). PSPSP comprises PSMPSP (20 features), PSDPSP (400 features) and 18 physicochemical properties. The different classes of amino acids corresponding to the physicochemical properties considered are charged (D, E, K, H, and R), aliphatic (I, L, and V), aromatic (F, H, W, and Y), polar (D, E, R, K, Q, and N), neutral (A, G, H, P, S, T, and Y), hydrophobic (C, F, I, L, M, V, and W), positively charged (K, R, and H), negatively charged (D and E), tiny (A, C, D, G, S, and T), small (E, H, I, L, K, M, N, P, Q, and V), large (F, R, W, and Y), transmembrane amino acid (I, L, V, A), dipole < 1.0 (A, G, V, I, L, F, P), $1.0 <$ dipole < 2.0 (Y, M, T, S), $2.0 <$ dipole < 3.0 (H, N, Q, W), dipole > 3.0 (R, K), and dipole > 3.0 with opposite orientation (D, E, C) [238, 277]. In order to calculate the physicochemical properties, the sum of the percentage composition of the amino acids belonging to each of these 18 classes is considered. These features altogether form the protein feature matrix $P_{n\times 438}$ for n protein sequences.

Position specific secondary structure profile (PSSSP): We have considered three types of secondary structures of a protein, i.e., helix (H), coil (C) and sheets (E) to form the matrix $SS_{n\times 3}$ for n protein sequences. The amino acids E, A, L, M, Q, K, R and H form helix in secondary structure format. Likewise, the amino acids G, N, P, S and D are known to form coil. Lastly, the amino acids V, I, Y, C, W, F and T collectively form sheet. In order to find the secondary structure composition, the sum of the percentage composition of the amino acids belonging to helix, coil and sheets are considered.

Presence of helices or coiled coils or sheets as domains in effector proteins has its own significance. Helices confer evolvability [49], attachment to host membrane [420], actin nucleation [105]. Crystal structures of the effector domains from two oomycete RXLR proteins, *Phytophthora capsici* AVR3a11 and *Phytophthora infestans* PexRD2 reveal a conserved core α-helical fold [49]. The fold exists in $\sim 44\%$ of the annotated *Phytophthora* RXLR effectors, both as a single domain and in tandem repeats of up to 11 units [49]. According to Boutemy *et al.*, the core α-helical fold displays the evolution of effector proteins to gain new virulence functions and/or evade the host immune system by insertion/deletions in loop regions between α-helices, extensions to the N and C termini, amino acid replacements in surface residues, tandem domain duplication, and oligomerization. A study by Weigele *et al.* [420] suggested that *Shigella* IpgB1 utilizes an amphipathic helix enriched with basic residues to interact directly with acidic phospholipids of host cell membrane. *Vibrio* T3 effector protein VopL contains three closely spaced WH2 domains (short 17-22 residues regions nearly always found in tandem and forming an N-terminal helix with a conserved downstream LKKV motif) which take part in actin stress fibre formation by directly nucleating actin filaments [105].

Coiled coil (alpha-helices coiled together) domains impart membrane attachment [221] and immunity [373]. Knodler *et al.* [221] suggested that coiled-coil domains are prevalent in virulence-associated proteins, including T3 effectors in *Salmonella enterica serovar Typhimurium*. These domains may represent a common membrane-targeting determinant for *Salmonella* T3 effectors [221]. Distinct regions of the *Pseudomonas syringae* coiled-coil effector AvrRps4 are required for activation of immunity [373]. Presence of β-sheets in ef-

Table 3.1: Summary of the distribution of amino acids based on their dipole and volumes of the side chains

Group	Amino acid
1	Alanine (A), Glycine (G), Valine (V)
2	Isoleucine (I), Leucine (L), Phenylalanine (F), Proline (P)
3	Tyrosine (Y), Methionine (M), Threonine (T), Threonine (S)
4	Histidine (H), Asparagine (N), Glutamine (Q), Tryptophan (W)
5	Arginine (R), Lysine (K)
6	Aspartic acid (D), Glutamic acid (E)
7	Cysteine (C)

fector proteins facilitate host-pathogen interaction [58]. *Salmonella* effector Protein SopB forms an inter-molecular β-sheet with Cdc42 of the host organism [58].

Position specific solvent accessibility profile (PSSAP): The solvent accessibility feature of an amino acid can be very buried (B), somewhat buried (b), very exposed (E), and somewhat exposed (e). We have considered these 4 features to form the solvent accessibility feature matrix $SA_{n\times4}$ for n protein sequences. The solvent accessibility has been calculated using the DSSP [205] program. An amino acid is said to be very buried (B) when its accessibility is at most 4%, somewhat buried (b) when accessibility is between 4% and 25%, somewhat exposed (e) when accessibility is between 25% and 50% and very exposed (E) when accessibility is more than 50% [287, 334]. Amino acids that can be characterized as very buried are A, L, F, C, I, V; somewhat buried amino acids are W, M, S, P, T, H and Y. Similarly, amino acids that are exposed are Q, E, D; and amino acids that are somewhat exposed are R, K, N and G. In order to calculate the solvent accessibility profile, the sum of the percentage composition of the amino acids being very buried, somewhat buried, very exposed and somewhat exposed are considered.

The solvent accessibility of a protein has an influence on their structure which in turn influences their functionality [414]. The extent to which the structure of proteins has an impact on their function is shown by the effect of changes in the structure of a protein. Any change to a protein at any structural level, including slight changes in the folding and shape of the protein, may render it non-functional [422]. The solvent accessibility feature of proteins is often used for identifying gram negative effector proteins [439, 440].

Conjoint Triad Descriptors (CTD): These features have been extracted from the amino acid sequences. The conjoint triad descriptors consider a group of three consecutive amino acids (triads) with respect to the protein sequences and their assigned groups depending on the classification based on dipole scale of each amino acid and volumes of side chains [81].

The distribution of the amino acids in each group has been given in Table 3.1. There are seven classes into which 20 amino acids can be placed. We have considered three consecutive amino acids (triplet) for further calculation. Considering three consecutive amino acids, each of the three amino acids will belong to one of the groups. The combination of the groups for three consecutive amino acids looks like $[3, 1, 7]$, for example, if these three amino acids are in Groups $3, 1$ and 7 respectively. Since three positions have been taken into consideration and each amino acid can belong to a single group, there can be one of 343 $(= 7 \times 7 \times 7)$ possible groups for each triplet of amino acids. The frequency of triplets belonging to each of these 343 combinations of groups are taken into account to obtain the final matrix of order $n \times 343$ (CTD), where n is the total number of sequences. The frequency of each triad belonging to one of the combinations of groups forms the CTD. For example, considering the peptide sequence $IMFTLED$. The combinations of IMF, MFT and FTL are $[2, 3, 2]$, $[3, 2, 3]$ and $[2, 3, 2]$. Hence, the frequencies of $[2, 3, 2]$, $[3, 2, 3]$, $[3, 2, 6]$ and $[2, 6, 6]$ are 2, 1, 1 and 1 respectively, while the rest of the groups have frequencies of 0.

The features $G_{n\times85}$, $P_{n\times438}$, $SS_{n\times3}$, $SA_{n\times4}$ and $CTD_{n\times343}$ have been combined to form a single feature matrix $F_{n\times(85+438+3+4+343=873)}$ as shown in Figure 3.2 for 873 features corresponding to each of the n genes/proteins. We have also generated some other features but could not consider them due to their non-conclusiveness. We have not included information regarding Pfam domains, palindrome sequences, nucleotide analysis of N and C terminals in our analysis due to their insignificant contribution [409] in distinguishing between effectors and non-effectors. We could not find any universal or major represented Pfam domain for the dataset taken from SecRet6 and SecretEPDB databases. We have not found any common palindrome sequence in the candidate genes. Moreover, we have considered amino acid composition, dipeptide composition and physicochemical properties of N and C terminals of the amino acid sequences, but could not find any significant differentiating factor between the effector and the non-effector proteins.

3.2.3 Secondary structure-based feature analysis of the effectors and non-effectors

The secondary structure composition of the effector and non-effector proteins displays contrast in distribution. A considerable difference has been noticed in the distribution of α-helices and β-sheets in both the categories of proteins. As given in Table 3.2, the overall percentage of helices in effector proteins is less than that in non-effectors. Similarly, the percentage of β sheets is more in effector proteins than in non-effectors. Statistical analysis of the correlation between α-helices and β-sheets in effectors shows a strong positive correlation among them with a p-value $<$ 0.05 and a Pearson correlation coefficient r = 0.88. Such a correlational significance is absent in non-effector proteins. Although we could not establish any immediate relevance of such a finding, the stark contrast in the distribution pattern of α-helices and β-sheets needs further investigation.

Table 3.2: Composition of secondary structures in the experimentally verified T6 effector proteins.

Class	Coil (in %)	Helix (in %)	Sheet (in %)
Effector proteins	45.48	19.69	33.42
Non-Effector proteins	39.11	40.98	18.43

3.2.4 Preprocessing of feature set

The cardinalities of the sets of effector and non-effector proteins are unbalanced, i.e., the number of samples in the effector class is considerably less than the number of samples in the non-effector class due to the unavailability of more experimentally verified T6 effector proteins. Equal sized sets of effector and non-effector proteins need to be considered to avoid unequal class distribution and a biased classifier [150]. In order to do so, we have over-sampled the training dataset using Borderline-SMOTE oversampling method [175] so that cardinality of the minority class (T6 effector proteins) has become approximately equal to that of the majority class (non-effector proteins). Borderline-SMOTE over-samples only the borderline examples of the minority class. For every minority sample, its k-nearest neighbors of the same class have been found, followed by the selection of some random samples from them according to the over-sampling rate. Hence, the new synthetic examples are generated along the boundary of the minority class and its selected nearest neighbors. This is followed by standardization of the features by subtracting them from the mean followed by scaling them to unit variance.

We have used Gini impurity index in a randomized decision tree [401] for feature selection on various sub-samples of the dataset to avoid over-fitting and improve the predictive accuracy of the classifiers. The classifiers have been tested on 10, 20, 30 . . .,850, 860, 873 features with 10 most significant features getting added in each iteration [113]. The performance of the predictors has been recorded for such datasets of different feature size and plotted in Figure 3.3 (a). It has been found that out of 873 features, 51 most significant features with respect to Gini impurity index are of high importance. The classifier has been seen to have relatively stable with negligible difference in accuracy. The size of the feature set has been increased from 51 most significant features achieving an $accuracy$ of 92.13%, to include all the 873 features resulting in an $accuracy$ of 95.36%. The variation of performances of the classifiers on dataset of different sizes has been depicted in Figure 3.3 (a). To avoid over-fitting, only these highly important features have been used for further classification and prediction.

3.2.5 Architecture of PyPredT6

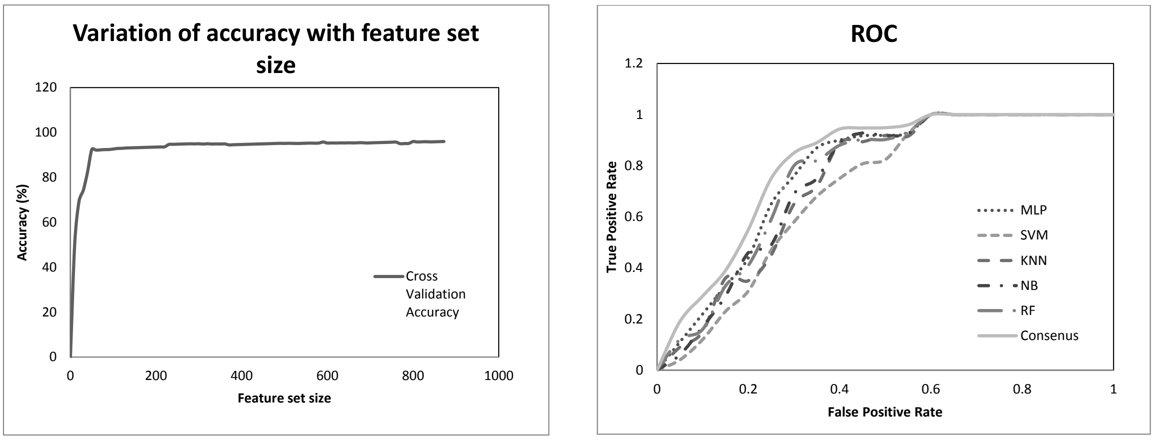
The performances of multiple models have been analyzed to derive a suitable model for the classification of T6 effectors. The models considered are support vector machine (SVM), k-nearest neighbor (k-NN), decision tree (DT), naive-Bayes (NB) and random forest (RF). Apart from them, an ensemble model consisting of these individual models has been considered. The result of classification by the ensemble model is decided via the strategy of majority voting. Since five classifiers have been considered, the issue of a tie among classifier predictions does not arise. A hard voting strategy, which is a type of majority voting, has been used to generate results from the individual classifiers. In hard voting, every individual classifier votes for a class, and the class with maximum votes wins. On analysis, an ensemble model with majority voting has been proposed for identification of T6 effectors. The performance of the ensemble model has been compared against each of the individual models considered. The ensemble model has reported a better performance compared to the individual models, as depicted in Table 3.3. Therefore, the ensemble model with majority voting has been chosen for developing PyPredT6.

3.3 Results

PyPredT6 uses the consensus of MLP, SVM, k-NN, NB and RF classifiers. It decides whether an unknown protein is a T6 effector or not using the method of majority voting. In this respect, the predicted values of all the five classifiers have been taken into consideration. The class predicted by majority of classifiers has been considered as the final class for a certain protein.

The MLP classifier has 6 hidden layers having the activation function ReLU for each node. The output layer nodes have the sigmoid activation function. The SVM classifier uses RBF kernel. The k-NN classifier has considered $k = 10$. The performance of the different classifiers has been assessed by $Accuracy, Sensitivity, Specificity, F - score$ and $G - mean$.

The individual performance and the consensus performance of the classifiers have been tabulated in Table 3.3. The Receiver Operating Characteristic (ROC) curve [177] for the same has been depicted in Figure 3.3 (b). As evident from the table and the plot, the consensus of the classifiers gives a better performance in identifying an unknown protein to be a T6



(a) Accuracy vs Feature set size

(b) ROC Curve

Figure 3.3: Performance of PyPredT6. (a)-represents the variation of accuracy with the feature set size. (b)-represents the ROC curve comparing the individual performances of the five classifiers and the consensus of classifiers. As visible, consensus of the five classifiers gives a better prediction result compared to the individual classifiers.

Table 3.3: Summary of performance (in %-age) of the five classifiers with 10-fold cross-validation. The tabulated values are the 50-fold average for each of the classifiers.

Classifier	$Accuracy$	$Sensitivity$	$Specificity$	F-score	G-mean
Multilayer perceptron	87.52	88.35	86.15	84.03	85.35
Support vector machine	84.57	80.80	87.70	81.54	87.24
k-nearest neighbors	88.05	81.82	87.25	82.56	84.69
Naive Bayes	89.42	81.27	88.45	85.42	84.63
Random forest	76.15	81.25	83.75	86.32	83.45
PyPredT6	**92.15**	**91.25**	**90.75**	**87.45**	**88.39**

effector or a non-effector.

3.3.1 Application of PyPredT6 on proteins of *Vibrio cholerae* and *Yersinia pestis*

The consensus of the five classifiers has been used to predict probable T6 effector proteins in *V. cholerae* and *Y. pestis*. The amino acid sequences for both the species have been obtained from Biocyc [211]. We have collected 2736 nucleotide and their respective amino acid sequences of *V. cholerae*. We have also collected 3850 nucleotide and their respective amino acid sequences of *Y. pestis*.

Out of 2736 proteins of *V. cholerae* (Chromosome 1: Strain O1 biovar El Tor str. N16961[4], version 21.1), 30 proteins have been selected by PyPredT6 to be probable effectors. For *Y. pestis* (Strain Pestoides F, version 21.1[5]), out of 3850 proteins, 42 proteins have been selected to be effectors. Here the predicted probable T6 effector proteins of our two test species have been discussed after secondary structure-based feature filtering. In order to biologically validate these proteins, we have considered their gene ontology information (as listed in UniProtKB[6] [90]) and information from existing literature. In this way, we have established a direct/indirect relation with virulence and pathogenesis of a few of these proteins. The literature-based validation of probable T6 effector proteins predicted by PyPredT6 has been furnished in the following two subsections. A tabulated form of the same has been furnished in `http://projectphd.droppages.com/PyPredT6.html`.

Predicted probable effector proteins in *Y.pestis* In *Y. pestis*, the list of 42 predicted probable T6 effector proteins include enzymes, flagellar proteins and auto-transporters, among others. Bacteriophage tail proteins are used by many pathogenic bacteria, for their secretion system and pathogenicity [208]. As demonstrated by Leiman *et al.* [246] and Pell *et al.* [313], bacteriophage tail protein can be an effector protein associated with T6SS of *Y. pestis*.

Bacterial PLA plays a crucial role in bacteria-induced haemolysis, thus providing a beneficial source of nutrients for bacterial growth in addition to providing an appropriate environment for survival and replication [199]. The pldA gene in *Y. pseudotuberculosis* encodes a phospholipase, the crucial phospholipid components of the outer leaflet of eukaryotic cell membranes. The 1468bp sequence, which includes the pldA gene with flanking regions, has been found to be 100% similar to the corresponding sequence of *Y. pestis* [210]. Proteases contribute to pathogenesis by affecting biological processes across the bacterial envelope.

Virulence of outer membrane usher protein (gene: caf1A), adhesin (gene: psaA), chaperone protein PsaB (gene: psaB), pesticin immunity protein (gene: pim), needle complex outer membrane lipoprotein precursor (gene: virG), outer membrane protein (gene: YopM) and target effector protein (gene: ypkA) have already been reported in [176]. Outer membrane usher protein, adhesin, and chaperone protein psaB are involved in pilus organization, assembly, and pathogenesis. Outer membrane lipoproteins are known to be required for T6SS in *E. coli* [18]. Effector protein product of ypkA gene has protein serine/threonine kinase activity. Among them, outer membrane and target effector proteins have also been reported as probable/putative effector proteins [176]. Lipoproteins contribute to the virulence of pathogens [394]. Chromosomal deletion of a lipoprotein gene sequence has resulted in

[4] `https://biocyc.org/VCHO/organism-summary?object=VCHO`

[5] `https://biocyc.org/YPES386656/organism-summary?object=YPES386656`

[6] `http://www.uniprot.org/`

a drastic reduction in virulence. It has been detected as effectors in both the test organisms considered. A tabulated form of the same has been furnished in file 'yp_bioanalysis.pdf' available at `http://projectphd.droppages.com/PyPredT6.html`.

Predicted probable effector proteins in *V. cholerae* In *V. cholerae*, the list of 30 probable T6 effector proteins includes cold shock proteins, enzymes, lipoproteins, flagellar proteins, and ribosomal proteins among others. Chitinases play a major role in pathogenesis. Multiple pathogens contain chitin coat, that acts as a shield from both the external and the internal environment. Many other pathogens attack the host using chitinase [174]. The pathogens use chitin containing structures for transmission and subsequent infection in the host. Flagellar proteins contribute to adhesion of the pathogen to the host and facilitate further processes [172]. The flg gene that results in the flagellar protein, is known to be linked to a virulence gene in *S. typhimurium* [65]. The flagellar protein hence hints towards contribution to pathogenicity, in accordance with our results.

Ribosomal proteins are essential components for the promotion of pathogenesis [411]. ToxR-activated gene A protein has been known to activate multiple virulence genes in *V. cholerae* [115]. A tabulated form of the same has been furnished in file 'vc_bioanalysis.pdf' available at
`http://projectphd.droppages.com/PyPredT6.html`.

3.4 Comparison of PyPredT6 with Bastion6

Bastion6 [409] is the only other tool that attempts to predict T6 effector proteins. However, multiple limitations have been observed in the tool. Bastion6 has extracted experimentally verified data from SecretEPDB while PyPredT6 has taken into consideration data from both SecReT6 and SecretEPDB. The training dataset of Bastion6 is imbalanced (consisting of 20 effector proteins and 200 non-effector protein samples). A low number of positive samples and a high number of features for prediction indicate Bastion6 as a probable over-fitted classifier. PyPredT6, on the other hand, has a positive set of 175 effectors and 1497 non-effectors. In order to do away with the problem of an imbalanced dataset, oversampling has been performed using borderline-SMOTE technique.

The non-effector samples (negative dataset) of Bastion6 have been taken from two sources - the non-effector proteins of Zou *et al.* [456] and those in *Vibrio parahaemolyticus*. The non-effector proteins from Zou *et al.* comprise those which are not T4 effectors. Due to multi-functional nature of prokaryotic genes [206], this may not be a safe approach. Proteins which are not T4 effectors may have an association with T6SS machinery. On the other hand, *Vibrio parahaemolyticus* is a pathogenic gram-negative bacteria [249]. Hence there arises a risk of considering an effector protein as a non-effector in the negative dataset. In order to avoid all these issues, PyPredT6 has taken the non-effector dataset, from an experimentally verified non-pathogenic organism.

Bastion6 has considered 1096 features in total with a sample size of 220. Given the high number of features and the limited number of samples, the dire need for feature selection and oversampling is noticed, which if not done, will lead to over-fitting [379]. The avoidance of using a feature selection and an oversampling method for Bastion6 indicates over-fitting [222]. Hence, PyPredT6 has incorporated feature selection along with an oversampled large training set with the intention to avoid over-fitting. PyPredT6 has used the most significant 51 features on an oversampled dataset of size 2994. For Bastion6, the values of $accuracy$ (94.3%), $sensitivity$ (100%) and $specificity$ (88%) have a considerable variance among the measurements, indicating quite an unstable performance. PyPredT6, however, displays a stable performance over $accuracy$ (89.15%), $sensitivity$ (91.25%) and $specificity$ (90.75%), with a considerably low variance. Such high variance in the above measurements for Bastion6 may indicate overfitting. Besides, Bastion6 has displayed a low specificity indicating a high number of false positives and low number of true negatives.

Bastion6 has considered the result of a single SVM classifier to predict the effector proteins, whereas PyPredT6 takes into account the prediction of five classifiers, and uses a voting method to obtain the final class label of the sample protein. An extensive study has been performed to measure the CPU time of PyPredT6. The summary of the study has been given in Table A.1 in Appendix A. CPU time for PyPredT6 on three random datasets containing 10, 20 and 30 sequences have been recorded. Here a single sequence refers to a pair of nucleotide and the corresponding amino acid sequences. For each set of sequences, the time needed for training PyPredT6 (T_T) with the feature set of experimentally verified effectors and the time required to extract features from unknown sequences (T_E) have been recorded. The average of total execution time ($T_S = T_E + T_T$) of PyPredT6 (5.24 minutes on a 32GB RAM, 64 bit Windows operating system) on the aforesaid three datasets is considerably less than that of Bastion6 (29.6 minutes on Bastion6 server). As observed from the table, the training time is nearly constant with an average of 314.61 seconds, while the average time for feature extraction for a single sequence is approximately 0.0751 seconds.

PyPredT6 is a standalone application, which can be downloaded from the website (http://projectphd.droppages.com/PyPredT6.html/). Bastion6 is restricted to process less than 500 sequences per job with amino acid count between 50 and 5000. PyPredT6, on the other hand, does not have any limit on the length or the number of sequences.

From a total of 6586 proteins of two species, we have predicted 72 effector proteins. PyPredT6 aims to reduce the true negatives while predicting the effector proteins. Bastion6 has considered 12 species for predicting the effector proteins. Among them, it has validated two proteins as probable effectors, while we have validated the possibility of all the predicted 72 proteins for being probable effector proteins. A summary of the comparison has been given in Table 3.4.

Table 3.4: Summary of the fundamental differences between PyPredT6 and Bastion6

Field	PyPredT6	Bastion6
Database	SecRet6 and SecretEPDB	SecretEPDB
Sample size	175 effectors, 1497 non-effectors	20 effectors, 200 non-effectors
Non-effector set	Entire genome of non-pathogenic gram-negative bacteria *Bacteriodes vulgatus* [173]	Proteins which are non-effectors with respect to T4 effectors, and from *Vibrio parahaemolyticus*, a pathogenic gram-negative bacteria
Feature types	Peptide and nucleotide features	Peptide features
Oversampling technique	borderline-SMOTE	None applied
Feature selection technique	Randomized decision tree using Gini impurity index	None applied, indicating overfitting
Classifier	Consensus of MLP, SVM, KNN, NB, RF	SVM
Execution time	5.24 minutes	29.60 minutes
Input constraints	Able to handle any number of sequences of any size	Unable to handle more than 500 sequences per job, length of each sequence between 50 and 5000.
Performance comparison	89.15% (Acc), 91.25% (Sen), 90.75% (Spe)	94.3% (Acc), 100% (Sen), 88% (Spe)

In order to assess and benchmark the performance of PyPredT6 and Bastion6, we have created three sets Set 1, Set 2 and Set 3, of independent non-overlapping effectors and non-effectors data extracted from the public databases and the literature, for comparing the predictive power of PyPredT6 and Bastion6. The first dataset, Set 1 (Table 3.5), has been constructed taking all the T6 effector proteins of *Edwardsiella tarda* from Genbank. The second set, Set 2 (Table 3.6), has been constructed using a handful of proteins of *Homo sapiens* obtained from Genbank, which cannot be effectors. The third dataset, Set 3 (Table 3.7), consists of T6 effector proteins accumulated from the literature. PyPredT6 has shown promising results while predicting effector and non-effector proteins from a pool of unknown proteins. Bastion6, on the other hand, has been unable to provide any conclusive result on classification of these proteins belonging to Set 1 and Set 2. For Set 3, Bastion6 has been able to predict 6 out of 10 proteins correctly, while PyPredT6 has been able to predict 9 out of 10 proteins correctly.

Table 3.5: Set 1 - Effector Dataset of *Edwardsiella tarda* obtained from Genbank [35]

Name	Class	PyPredT6	Bastion6
evpP	Effector	Effector	Undefined
ImpC	Effector	Effector	Undefined
Hcp	Effector	Effector	Undefined
evpD	Effector	Effector	Undefined
ImpF	Effector	Effector	Undefined
ImpG	Effector	Effector	Undefined
ImpH	Effector	Effector	Undefined
VasG	Effector	Effector	Undefined
VgrG	Effector	Effector	Undefined
evpJ	Effector	Effector	Undefined
ImpA	Effector	Effector	Undefined
evpL	Effector	Effector	Undefined
ImpJ	Effector	Effector	Undefined
ImpK	Effector	Effector	Undefined
ImpL	Effector	Effector	Undefined
avtA	Effector	Effector	Undefined
transposase (A)	Effector	Effector	Undefined
wabN	Effector	Effector	Undefined
hemY	Effector	Effector	Undefined
lysC	Effector	Effector	Undefined

3.5 Conclusions

Prediction of effector proteins from bacterial genome information is important for the analysis of their secretion systems' role in pathogenesis. Here we have developed a standalone system, called PyPredT6, for prediction of probable T6 effector proteins based on consensus of five classifiers. PyPredT6 extracts a feature set having 873 features from nucleotide and amino acid sequences of experimentally verified T6 effector proteins. PyPredT6 has predicted 42 proteins out of 3850 proteins from *Y. pestis* and 30 proteins out of 2736 proteins from *V. cholerae* as effectors. We have analyzed these proteins for being putative T6 effector proteins in a limited capacity. PyPredT6 offers users to check whether a protein is a T6 effector or not. A more detailed biological validation for each putative candidate gene is essential, which forms a scope for further study. The methodology can be extended to other pathogens, whose genomes and proteomes are either partially or fully mapped.

Table 3.6: Set 2 - Non-effector dataset of *Homo sapiens* obtained from Genbank [35]

Name	Class	PyPredT6	Bastion6
cyclin-Y-like protein 3	Non-effector	Non-effector	Undefined
acidic phospholipase	Non-effector	Effector	Undefined
CPHXL	Non-effector	Non-effector	Undefined
SH3	Non-effector	Non-effector	Undefined
DUXB	Non-effector	Non-effector	Undefined
KRTAP9	Non-effector	Non-effector	Undefined
KRTAP16	Non-effector	Effector	Undefined
FTMT	Non-effector	Non-effector	Undefined
TRAM1L1	Non-effector	Non-effector	Undefined
CLDN17	Non-effector	Non-effector	Undefined

While working on this investigation, it came into our knowledge that no classification based investigation has been done taking into account the tertiary structure of these effector proteins. All of the investigations regarding effector proteins have been done on the basis of primary and secondary structure of the same. Keeping that in mind, we have come up with a unique system called Effector Protein Predictor based on 3D structure (EPP3D), to identify effector proteins based on their 3D structure. Since our effector dataset is not balanced and the available balancing techniques are not applicable to our dataset, we have also come up with a new oversampling technique called Cluster Quality based Non-Reductional (CQNR) oversampling technique. The development of EPP3D and CQNR has been described in Chapter 4.

Table 3.7: Set 3 - Dataset consisting of T6 effector proteins from various organisms. The tag "removed" is for those T6 effector proteins which were similar to one of the 175 proteins used in the training dataset.

Name	Organism	Reference	Class	PyPredT6	Bastion6
Hcp	*Desulfobacterium autotrophicum*	Wang *et al.*, 2015 [410]	Effector	Non-effector	Non-effector
TssA	*Pseudomonas fluorescens*	Durand *et al.*, 2014 [122]	Effector	Effector	Non-effector
Hcp-ET1 (removed)	*Escherichia coli*	Ma *et al.*, 2017 [265]	Effector	-	-
TssE	*Pseudomonas fluorescens*	Durand *et al.*, 2014 [122]	Effector	Effector	Non-effector
Hcp-ET3 (removed)	*Escherichia coli*	Ma *et al.*, 2017 [265]	Effector	-	-
VgrG3	*Pseudomonas fluorescens*	Durand *et al.*, 2014 [122]	Effector	Effector	Effector
Hcp-ET2 (removed)	*Escherichia coli*	Ma *et al.*, 2017 [265]	Effector	-	-
TssG	*Pseudomonas fluorescens*	Durand *et al.*, 2014 [122]	Effector	Effector	Effector
Hcp3	*Pseudomonas fluorescens*	Brunet *et al.*, 2015 [54]	Effector	Effector	Effector
EvpP	*Edwardsiella tarda*	Durand *et al.*, 2014 [122]	Effector	Effector	Effector
TecA	*Burkholderia cenocepacia*	Speiwak *et al.*, 2019 [20, 375]	Effector	Effector	Non-effector
Tse1	*Pseudomonas Aeruginousa*	Durand *et al.*, 2014 [122]	Effector	Effector	Effector
Tae4	*Enterobacter Cloacae*	Durand *et al.*, 2014 [122]	Effector	Effector	Effector
Hcp-ET5 (removed)	*Escherichia coli*	Ma *et al.*, 2017 [265]	Effector	-	-

Chapter 4

Cluster Quality-based Non-Reductional (CQNR) Oversampling Technique and Effector Protein Predictor Based on 3D Structure (EPP3D) of Proteins [355]

4.1 Introduction

In Chapter 3, we have developed a system, called PyPredT6, to identify T6 effector proteins based on their primary and secondary structures. However, no attempts have been made to predict effector proteins from their tertiary structures. Prediction of effector proteins would remain incomplete if their tertiary structure is not taken into consideration. Thus, in this chapter, we have developed a system for identification of effector proteins based on the characteristics of their tertiary structures.

Effector proteins in gram-negative bacteria are translocated into host cells predominantly by T3SS, T4SS and T6SS [337] [446] [32] [439]. T3SS has extensively been studied [148] [312]. Pathogens with T3SS effectors can infect both plants and animals [72] [148] [312] [439] [147]. T4SS, discovered in 1990s [234], is considered as one of the most functionally diverse bacterial secretion systems, both in terms of transported substrates and targeted recipients [70]. The working mechanism of T6SS has been discovered in 2006 [319]. Several aspects of the working mechanism of T6SS are still unknown. T3SS, T4SS, and T6SS associated effector proteins in many gram-negative bacteria are yet to be discovered. Several methods have been developed to classify/predict/identify T3, T4, T6 effector proteins based on their amino acid sequences [59, 353, 354, 406, 408, 409, 432, 433, 440, 456]. However, no 3D structural feature-based classification mechanism for differentiating T3, T4, T6 effectors (pathogenic proteins), other secretion system effectors, and non-pathogenic proteins have been reported so far in the literature [354].

Effector protein datasets tend to be highly imbalanced, due to the limited availability of effectors and the abundance of non-effectors. A dataset is said to be imbalanced if the number of samples belonging to each of the classes is unequal [77]. Imbalanced datasets pose a severe problem for decision making [323]. A classifier is biased towards the class having larger number of samples, and thereby yielding unsatisfactory performance [323]. Training a classifier with an imbalanced dataset leads to overfitting [71]. Inadequate predictions using biased and overfitted classifiers have led to the notion of balancing an imbalanced dataset. A well-balanced dataset is essential for designing a reliable classification and prediction model. For a 2-class classification problem, the class having the higher cardinality is called the majority class while the other class is referred to as the minority class. Data imbalance can be tackled either by eliminating the samples from the majority class (undersampling) or increasing the number of samples in the minority class (oversampling) [77]. No matter how varied the strategies for sampling are, both the sampling techniques aim at making the cardinalities of the classes equal.

One of the simplest algorithms for oversampling is the random oversampling technique [232]. Several other algorithms have been developed over time. They include SMOTE [77], borderline-SMOTE [175], C-SMOTE [181] and Safe-Level-SMOTE [57] among others. Zhang *et al.* [452, 453] have explored the task of balancing imbalanced image datasets for pathological brain detection. However, none of the oversampling techniques have taken any measure to regulate the generation of synthetic samples of the minority class, which may fall in the vicinity of majority samples. Moreover, some of the techniques discard samples as noise. However, discarding samples as noise may lead to loss of information embedded in the dataset. While some are incapable of handling multi-class imbalanced datasets, others are unable to handle high-dimensional data.

In order to overcome the shortcomings of the aforesaid oversampling algorithms and to identify effector proteins based on their 3D structures, we develop two algorithms in this chapter, one for oversampling and the other for identification of effector proteins based on their 3D structures. The chapter primarily has five parts. The first part describes the extraction of numerous features based on 3D coordinates of each atom of the experimentally verified effector proteins to form the feature set. However, the effector dataset is imbalanced, and the state-of-the-art algorithms do not generate satisfactory prediction results. Thus, in the second part of the chapter, we introduce a novel oversampling algorithm, called Cluster Quality-based Non-Reductional (CQNR) oversampling. CQNR has resulted in a significant improvement in performance over some existing oversampling techniques on benchmark datasets. In the third part, we develop a supervised learning-based system, called Effector Protein Predictor based on 3D structure (EPP3D) of proteins. EPP3D predicts the class of an unknown protein, after being trained with a balanced dataset obtained as the output of CQNR taking the imbalanced effector dataset as its input. EPP3D classifies unknown proteins into five classes, namely, individual classes of T3, T4 and T6 effector proteins, a composite class of T1, T2, T5, T7 effector proteins, and a class of non-effector proteins. The effectiveness of 3D structure-based classification of effector proteins has been exhibited using five classifiers individually as well as EPP3D. The performance comparison of CQNR and EPP3D against the state-of-the-art algorithms form the fourth part of the chapter. Finally, the chapter provides a qualitative discussion regarding the comparison.

4.2 Methodology

This section presents a vivid description of development of the oversampling algorithm CQNR and the effector identification system EPP3D. The first step towards the development of EPP3D is the data collection phase. Tertiary structure of a protein is of great significance since they give an insight into the shape of a protein, which dictates its functionality. Thus, tertiary structures of experimentally verified effector proteins have been collected depending on the availability of their 3D structures in the form of PDB files (Appendix B.2). For effector identification, we consider the following classes of effector proteins.

- Class 1 - T3 effector proteins
- Class 2 - T4 effector proteins
- Class 3 - T6 effector proteins
- Class 4 - Other (T1, T2, T7) effector proteins
- Class 5 - Non-effector proteins

The structure of a protein determines its function. In order to capture the characteristics of the tertiary structure, feature extraction has been carried out. From the tertiary structure of effectors and non-effectors, eight unique features have been extracted. These features represent various structural aspects of a protein. These aspects include binding capability, solvent accessibility and hydrophobicity of proteins. The extracted features have been integrated to form a feature set using which EPP3D has been designed. Due to an imbalanced training dataset and the poor performance of state-of-the-art oversampling techniques, we have developed a new oversampling algorithm, called CQNR. The effector dataset, oversampled by CQNR, has been used for development of EPP3D. It has been noticed that all the eight features are crucial enough, and have not been discarded by feature selection. Therefore, all the features have been taken into consideration for designing EPP3D.

To design EPP3D, an ensemble classifier with majority voting has been implemented. The ensemble system consists of five classifiers *viz.*, multi-layer perceptron (MLP), support vector machine (SVM), k-nearest neighbor (kNN), naive bayes (NB) and random forest (RF) classifiers. Cross-validation has been implemented for parameter tuning of individual classifiers in the ensemble model. EPP3D has been subjected to diligent training and testing methods for accurate identification of effector proteins. CQNR and EPP3D are standalone applications, which can be downloaded from the website `http://projectphd.droppages.com/CQNR.html/`.

4.2.1 Data collection

We have accumulated experimentally verified data associated with effector proteins of T3, T4, and T6 secretion systems in 35 different species, provided in Table 4.1, from different repositories/literature. These species are pathogenic to various living organisms, such as fish, amphibians, reptiles, birds, human, other animals, and various plants. We have accumulated information on 3D structures of T3, T4 and T6 effector proteins from databases, such as SecretEPDB [14], SecReT4 [40], SecReT6 [253] and Protein Data Bank [36].

Out of 1230 T3 effector proteins reported by SecretEPDB and 56 T3 effector proteins listed by Yang *et al.* [440], PDB structures of 36 effector proteins have been collected. Among 731 T4 effector proteins published in SecretEPDB [14] and 186 in SecReT4 database [40], PDB structures of 80 proteins have been found. Likewise, out of 107 T6 effector proteins summarized in SecReT6 database [253] and 181 in SecretEPDB, 31 PDB structures of T6 effector proteins have been obtained. Consolidating these data and removing redundancy, the summary of the ultimate list obtained has been provided in Table 4.2. No database containing information regarding T1, T2, T7, and non-effectors have been reported so far.

For the other groups of secreted proteins, i.e., T1, T2, T5, and T7, we have searched PDB to retrieve secreted proteins of different secretion systems[1]. The ”Others” class consisted of 2 T1SS, 19 T2SS, and 3 T7SS proteins. We could not find the 3D structure of any T5 effector protein. Due to the inadequacy of T1, T2 and T7 effectors, these effector proteins have been grouped into a single class, i.e., class 4, consisting of 24 proteins.

For the non-effectors, we have chosen two organisms, namely, *Bacteroides vulgatus* [173] and *Listeria innocua* [427], which are non-pathogenic. It may be mentioned here that there is no protein in pathogenic organisms, which has been experimentally verified to be non-effectors. Here, an argument may arise that the housekeeping proteins of the same pathogenic species might have been considered as the non-effector proteins. However, in prokaryotes, genes are often found to be multi-functional [206, 314, 428]. Furthermore, a housekeeping protein of the same species may have a direct or indirect association with an effector protein [341]. Therefore, we have considered the proteins of two non-pathogenic organisms as non-effectors. We have collected 120 proteins of experimentally verified non-

[1] PDB has effector protein names with “T*SS” in them, where “*” denotes the secretion system type 1, 2, 3, 4, 5, 6 or 7.

Table 4.1: Cardinality of the dataset. The number of effector proteins collected from each species is given in parenthesis alongside it.

Secretion System	Species	Pathogenicity
T1SS (1 species & 2 proteins)	*Serratia marcescens*	Human
T2SS (9 species &19 proteins)	*Pseudomonas aeruginosa* (5)	Human
	Escherichia coli (3)	Human
	Vibrio cholerae (5)	Human
	Aeromonas hydrophila (1)	Human
	Dickeya dadantii (1)	Plant
	Klebsiella pneumoniae (2)	Human
	Vibrio vulnificus (1)	Human and animal
	Vibrio parahaemolyticus (1)	Human
T3SS (1 species & 56 proteins)	*Pseudomonas syringae* (56)	Plant
T4SS (10 species & 186 proteins)	*Helicobacter pylori* (7)	Human stomach
	Agrobacterium tumefaciens (5)	Plant
	Brucella melitensis (4)	Animals and Human
	Coxiella burnetii (25)	Animals and Human
	Bordetella pertussis (5)	Human
	Bartonella henselae (7)	Human
	Legionella pneumophila (123)	Human
	Anaplasma marginale (1)	Livestock animals
	Brucella melitensis (4)	Livestock animals and human
	Agrobacterium rhizogenes (5)	Plant
T6SS (26 species & 87 proteins)	*Yersinia pseudotuberculosis* (2)	Animals and human
	Yersinia pestis (1)	Animals and human
	Vibrio cholerae (7)	Human
	Vibrio alginolyticus (1)	Human
	Serratia marcescens (1)	Human
	Pseudomonas syringae (1)	Plant
	Pseudomonas protegens (2)	Plant protecting bacteria (non-pathogenic)
	Pseudomonas aeruginosa (17)	Plants, animals, human
	Paracoccus denitrificans (1)	Denitrifying bacteria (non-pathogenic)
	Helicobacter hepaticus (2)	Mice, human
	Francisella tularensis (7)	Bird, reptile, fish, animals, human
	Escherichia coli (5)	Intestine of animals, human (mostly non-pathogenic)
	Erwinia carotovora (5)	Plant
	Enterobacter cloacae (1)	Human
	Edwardsiella tarda (3)	Fish, amphibians, reptiles, mammals
	Citrobacter rodentium (1)	Mice
	Burkholderia thailandensis (14)	Animals, human
	Burkholderia pseudomallei (2)	Animals, human
	Burkholderia mallei (3)	Animal
	Burkholderia cenocepacia (1)	Plants, human
	Agrobacterium fabrum (2)	Plant
	Aeromonas hydrophila (4)	Fish, amphibians, human
	Acinetobacter baumannii (1)	Human
	Flavobacterium johnsoniae (1)	Fish
T7 (1 species & 3 proteins)	*Streptococcus intermedius*	Human

Table 4.2: Summary of the data comprising T1, T2, T3, T4, T6, T7 effector proteins, and non-effector proteins considered. The column ”Databases” contains the number of a particular class of effector or non-effector proteins obtained from a particular database whose reference has been given within ”()”. The column ”PDB” indicates the number of effector or non-effector proteins of a particular class, obtained from Protein Data Bank. The structures of proteins from the respective databases, whose 3D structures are found in PDB, have been used to create the final set of experimentally verified proteins for training the classifiers of EPP3D.

Secretion system	Number of proteins from	
	Databases	PDB Structure
T1	-	2
T2	-	19
T3	56 (Yang *et.al.*), 1230 (SecretEPDB)	36
T4	186 (SecReT4), 731 (SecretEPDB)	80
T5	No data found	No data found
T6	107 (SecReT6), 181 (SecretEPDB)	31
T7	-	3
Non-effector	-	120

pathogenic organisms to constitute the non-effector set.

4.2.2 Feature extraction

Effector proteins T3, T4, and T6 bind with host proteins. The binding alters the working mechanism of these host proteins, which eventually disrupts the regular function of the host [166]. Thus, understanding 3D structural characteristics of pathogenic effector proteins is indeed crucial for exploring the mechanism of protein-protein interactions between host proteins and effector proteins [47]. Three-dimensional structural characteristics of pathogenic effectors (T3, T4, T6), effectors from other secretion systems (T1, T2, T7) and proteins from non-pathogenic organisms have been characterized in terms of eight features. A detailed description of these features based on 3D structures of the proteins along with their significance, is furnished below.

A protein structure can be perceived as a point cloud, where a point corresponds to an atom, a constituent element of the protein in the 3D coordinate system. The concept of a point cloud has been utilized to generate the values of eight features, namely, radius of gyration (r_g), compactness (f), convex hull layer count (h), surface atom composition (c_N - percentage composition of nitrogen atom, c_O - percentage composition of oxygen atom, c_S - percentage composition of sulfur atom, c_C - percentage composition of carbon atom) and packing density (d). These features provide essential information pertaining to the overall surface, packing pattern of atoms as well as hydrophilicity/hydrophobicity of surface atoms of a protein. These features additionally determine the binding potential of a protein molecule with other protein molecules [47]. Among them, h, f, c_N, c_O, c_S and c_C have been calculated using the notion of convex hull [380].

Radius of gyration (r_g). The radius of gyration of a protein is defined as the average distance between the center of a protein and each of its atoms [96]. The center of a protein is given by

$$(\overline{x}, \overline{y}, \overline{z}) \equiv \left(\frac{\sum\limits_{i=1}^{n'} x_i}{n'}, \frac{\sum\limits_{i=1}^{n'} y_i}{n'}, \frac{\sum\limits_{i=1}^{n'} z_i}{n'} \right), \tag{4.1}$$

where (x_i, y_i, z_i) is the coordinates of the ith atom of a protein containing n' atoms. Now, the radius of gyration of a protein is defined as

$$r_g = \frac{1}{n'} \sum_{i=1}^{n'} [(x_i - \overline{x})^2 + (y_i - \overline{y})^2 + (z_i - \overline{z})^2]^{\frac{1}{2}} \tag{4.2}$$

The term r_g provides a quantitative estimate of the size of a protein. Larger the value of r_g, larger is the size of the protein. A protein molecule with a large surface area is exposed to many binding sites of another protein molecule [106]. Hence, larger the size of the protein molecule, higher is the chance of binding.

Compactness (f). Compactness of a protein has been defined as a measure of molecular surface area [447]. Mathematically, it can be represented by the ratio of its accessible surface area to the surface area of a sphere having radius r_g [447]. In order to obtain the accessible area of a protein, convex hull of the protein is determined, which is formed by using the points corresponding to the atoms of the protein.

A convex hull of a set S of points is the smallest convex polyhedron that incorporates these points. The polyhedron is such that some of the points lie on the bounding surface while the others are inside it. The convex hull of a protein has been obtained by using the quickhull algorithm [30]. Let $Conv(S)$ be the set of points on the bounding surface of the convex polyhedron. A triangulation of a finite point set $S \subset \mathbb{R}^3$ is a set $\mathscr{T}$ of triangles such that:

- $Conv(S) = \bigcup\limits_{j} V(T'_j)$ where $V(T'_j)$ is the set of vertices forming j^{th} triangle T'_j.
- For every distinct pair $T'_j, T'_{j'}, \in \mathscr{T}$, T'_j and $T'_{j'}$ have either a common vertex, a common edge or none.

Here $T' =< \mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3 >$, such that $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$ are the points forming the triangle T'. The vector $\mathbf{t}_i$ is represented by the coordinates (x_i, y_i, z_i). Thus compactness (f) of a protein is computed as

$$f = \frac{\sum\limits_{j=1}^{p} \Delta(T'_j)}{4\pi r_g^2} \tag{4.3}$$

where $\Delta(T'_j)$ denotes the area of jth triangle T'_j forming a part of the convex hull, p is the number of such triangles formed, and $\sum\limits_{j=1}^{p} \Delta(T'_j)$ denotes an estimate of the accessible surface area of the protein. The utility of this measure is to identify protein domains. The functionality of a protein depends on its domain. Protein domain regions, compared to other regions of the protein, are more compact. Due to protein folding, reduction in the surface area is linearly dependent on hydrophobicity [158]. Lower the surface area, lower is the hydrophobicity of a protein.

Convex hull layer count (h). The convex hull of a protein has been determined by using the quickhull algorithm [30] as defined above. The convex hull of a protein corresponds to its solvent accessible surface area [89]. Initially, a convex hull is formed, considering all the atoms of a protein. Then the points on the convex hull are eliminated. The next convex hull is obtained afresh using the remaining set of atoms, and the points on the convex hull are removed. This process continues till there is no more point left [94]. If S is the set of all the points representing atoms in a protein, the next set of vertices on which convex hull will be formed, is

$$\begin{gathered} S_1 = S - Conv(S), \\ S_2 = S_1 - Conv(S_1), \\ . \\ . \\ . \\ S_h = S_{h-1} - Conv(S_{h-1}) \\ S_{h+1} = \varnothing \end{gathered} \tag{4.4}$$

where h is the number of convex hull layers, and $Conv(S)$ is the function that returns the set of points lying on the convex hull obtained from S as shown in Figure 4.1. The term h is called the convex hull layer count. As mentioned above, the convex hull layer provides information concerning the solvent accessibility analysis for proteins [380]. Convex hull layer count h provides a physical distance through which a solvent molecule would have to travel to reach the core of the protein from its surface, thus giving us an insight into how deep a protein molecule is. This feature determines the binding surface of a protein with another protein [78]. Convex hull gives a measure of the exposed surface area of a protein, hence giving an insight into its available binding area.

Surface atom composition. Investigation of the surface atoms of a protein presents an

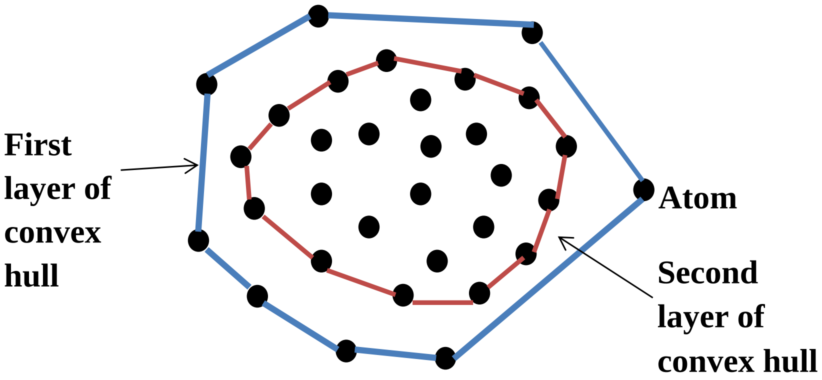


Figure 4.1: Schematic diagram depicting the formation of convex hull layers. A point cloud has been taken into consideration. The atoms of a protein are denoted by these points (black) in the point cloud. The outer boundary (depicted in blue) is the first convex hull layer created with the surface atoms of an effector protein. The inner boundary (depicted in red) is the second convex hull layer created in a similar manner with the surface atoms after removing the atoms on the first convex hull layer.

insight into the estimation of hydrophobic forces and their subsequent effect on protein structure [182,331]. We have extracted the percentage composition of nitrogen (c_N), carbon (c_C), oxygen (c_O) and sulfur (c_S) atoms present on the first convex hull layer of the experimentally verified effectors and non-effectors. Thus we have got four features.

Packing Density (d). It measures the packing pattern of atoms in a protein [358]. Packing Density (d) is defined as the ratio of total volume of all the atoms to the volume of protein, and is given by,

$$d = \frac{\sum_{i=1}^{n'} v_i}{\frac{4}{3}\pi r_g^3} \tag{4.5}$$

where v_i is the volume of ith atom [447] and r_g is the radius of gyration of the protein. The volume of each atom has been estimated using the radius of nitrogen (N), oxygen (O), carbon (C) and sulfur (S) atoms available from the database. Packing density has a pronounced effect on the binding property of the protein [247].

The effector protein dataset that we consider in this chapter, comprises experimentally verified 36 T3, 80 T4, 31 T6, 24 effectors of T1, T2 and T7, and 120 non-effector proteins from two non-pathogenic bacteria - *Bacteriodes vulgatus* and *Listeria innocua*. Each of these proteins is characterized by the above eight features extracted from its 3D structure. The dataset is clearly an imbalanced one. In order to balance the same, we develop CQNR for oversampling the dataset. The working mechanism of CQNR has been furnished below.

4.2.3 Cluster Quality-based Non-Reductional (CQNR) oversampling technique

The fundamental objective of algorithm CQNR is as follows: finding the best number of clusters using Davies-Bouldin index [98] for the samples of the minority class(es) to be clustered using K-means clustering algorithm, followed by generating well-spaced points within a cluster in proportion to the size of each cluster. We have applied K-means as it is widely used and produces tighter clusters compared to other clustering methods [405]. It has been found that the performance of Davies-Bouldin index in identifying the appropriate number of clusters is better compared to many other cluster validity indices [154]. The comparative study regarding clustering algorithms and cluster validity indices has been reported in Table A.2 of Appendix A. A summary of the variables used in the algorithm has been given in Table 4.3.

Algorithm 1 Cluster Quality-based Non-Reductional (CQNR) oversampling technique

```
Procedure $CQNR(\mathscr{C}_1, \mathscr{C}_2)$
Check the cardinalities of the classes $\mathscr{C}_1, \mathscr{C}_2$
Assign to $\mathscr{L}$ the class with the larger cardinality and to $\mathscr{S}$ the other class
Find $D_g$ for $g$ clusters obtained from $\mathscr{S}$ for $2 \leq g \leq 20$
Assign to $m'$ the value of $g$ for which minimum $D$ score is obtained
Find the difference between the cardinalities of $\mathscr{L}$ and $\mathscr{S}$
Calculate $\theta_l$ using equation 7
$\mathscr{N} = \varnothing$
for $l = 1 : m'$
    $q = 1$
    while $q \leq \theta_l$
        Select 2 random data samples $(\mathbf{a}_1, \mathbf{a}_2)$ from $l$th cluster $C_l$
        Generate a random $w_1$ in $(0, 1)$
        $w_2 = 1 - w_1$
        Generate a synthetic sample $\mathbf{b'}_{lq}$ such that
        $\mathbf{b'}_{lq} \leftarrow w_1\mathbf{a}_1 + w_2\mathbf{a}_2$
        Calculate $d'$, $R_l$ using equations 10 and 12
        if $d' \leq R_l$ then
            Put $\mathbf{b'}_{lq}$ to $\mathscr{N}$
            $q = q + 1$
$\mathscr{N} = \mathscr{S} \cup \mathscr{N}$
return $\mathscr{N}$
```

Let us consider a two-class classification problem where the associated dataset is imbalanced. Let $\mathscr{S}$ be the minority class and $\mathscr{L}$ be the majority class. Each sample of majority or minority class is defined as $\mathbf{b} = [f'_1, f'_2, \ldots, f'_\mu]^T$ where f'_e is the feature value and μ is the number of features and $1 \leq e \leq \mu$. The required number of synthetic data samples to be generated is

$$\delta = \zeta_1 - \zeta_2, \tag{4.6}$$

where ζ_1, ζ_2 are the cardinalities of the classes $\mathscr{L}$ and $\mathscr{S}$ respectively. CQNR is applied to the minority class $\mathscr{S}$. Initially, all samples in the minority class have been taken into consideration. K-means clustering algorithm has been applied to the samples in the minority class. The number of clusters generated ranges from 2 to 20. The most suitable number (m') of clusters in the minority class is obtained by Davis-Bouldin index. CQNR takes this m' value for further operations, as it has turned out to be the best with respect to Davies-Bouldin Index. Further, the cardinalities of these clusters sum up to the cardinality of the minority dataset. We now aim at generating synthetic samples in such a way that the percentage contribution of each cluster to the cardinality of the entire minority class is sustained. Let θ_l be the number of synthetic data samples that need to be generated in l^{th} cluster C_l of the minority class. Then,

$$\theta_l = \left\lfloor \frac{|C_l|}{\zeta_2} \times \delta \ + 0.5 \right\rfloor, 1 \leq l \leq m' \tag{4.7}$$

where

$$\delta \approx \sum_{l=1}^{m'} \theta_l \tag{4.8}$$

To balance the dataset following the original distribution of cluster C_l, CQNR generates a new synthetic sample as a weighted sum of the randomly selected minority class samples. It might so happen that among two random samples selected from the minority class, one of the samples may have feature values close to that of samples belonging to the majority class. For such a case, the weighted sum of these two randomly chosen minority class samples subsequently may fall in the region of the majority class. In order to avoid the generation of synthetic samples in the region of majority class, the radius of a cluster has been considered. The center of the cluster is calculated to obtain the radius of the cluster. The distance of a newly generated synthetic point for a particular cluster from the cluster center is calculated and reviewed if the distance is less than the radius. If the distance is less than the radius, the synthetic sample is retained, else it is discarded.

For generation of θ_l synthetic samples, weighted sum of two randomly selected samples from C_l is considered. The cluster center $\overline{\mathbf{b}}_l$ and the radius of the cluster R_l are given by,

$$\overline{\mathbf{b}}_l = \frac{1}{|C_l|} \sum_{k=1}^{|C_l|} \mathbf{b}_{lk} \tag{4.9}$$

$$R_l = \frac{1}{|C_l|} \sum_{k=1}^{|C_l|} ||\mathbf{b}_{lk} - \overline{\mathbf{b}}_l|| \tag{4.10}$$

Table 4.3: Summary of the variables used in this chapter

Name	Description
h	convex hull layer count
c_N, c_C, c_O, c_S	count of nitrogen, carbon, oxygen and sulfur atoms on the surface of the molecule
r_g	radius of gyration of a protein molecule
n'	number of atoms in a protein molecule
d	packing density of a protein molecule
x_i, y_i, z_i	the x,y,z coordinates of the ith atoms in a protein molecule, $1 \leq i \leq n$
p	number of triangles forming the convex hull
T'_j	set of points forming the jth triangle of convex hull, $1 \leq j \leq p$
$\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$	the three points forming T'_j
$\mathscr{L}$	majority class
$\mathscr{S}$	minority class
μ	number of features in the dataset
f'_e	eth feature of the dataset, $1 \leq e \leq \mu$
ζ_1	number of samples in the majority class
ζ_2	number of samples in the minority class
δ	difference in the sample size of the classes
m'	number of clusters to be considered
C_l	lth cluster on k-mean clustering, $1 \leq l \leq m'$
$\mathbf{b}_{lk}$	kth sample of the lth cluster of minority class, $1 \leq l \leq m, 1 \leq k \leq \|C_l\|$
$\overline{\mathbf{b}}_l$	center of the lth cluster, $1 \leq l \leq m'$
d'	distance of $\mathbf{b}_{lk}$ with the center $\overline{\mathbf{b}}_l$ of the lth cluster, $1 \leq l \leq m, 1 \leq k \leq \|C_l\|$
θ_l	number of synthetic samples to be generated in C_l, $1 \leq l \leq m'$
D_g	Davies-Bouldin index for g clusters, $2 \leq g \leq 20$
$\mathbf{b}'_{lq}$	a newly generated qth synthetic sample of lth cluster, $1 \leq l \leq m'$, $1 \leq q \leq \theta_l$
R_l	radius the lth cluster, $1 \leq l \leq m'$
w_1, w_2	random weight values to be generated, $0 < w_1, w_2 < 1$
$\mathbf{a}_1, \mathbf{a}_2$	random samples selected from a cluster of samples
$\mathscr{N}$	final balanced minority class dataset

Each synthetic sample $\mathbf{b'}_{lq}$ in C_l to be generated is given by

$$\mathbf{b'}_{lq} = w_1\mathbf{a}_1 + w_2\mathbf{a}_2, q = 1, 2, .., \theta_l; 0 < w_1, w_2 < 1, w_2 = 1 - w_1 \tag{4.11}$$

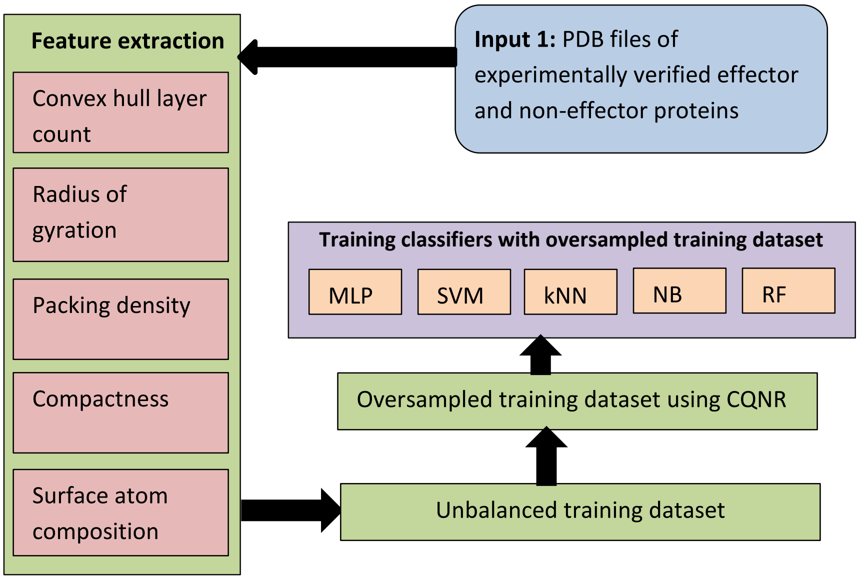
where $\mathbf{a}_1, \mathbf{a}_2$ are the two random samples selected from C_l, and w_1 is a random number generated in (0,1). The distance (d') of the newly generated sample $\mathbf{b'}_{lq}$ from the center $\overline{\mathbf{b}}_l$ is calculated. If $d' \leq R_l$, $\mathbf{b'}_{lq}$ is selected as a synthetic sample, else it is discarded, and another $\mathbf{b'}_{lq}$ is generated and checked for the criterion mentioned above. CQNR keeps generating distinct synthetic samples for which the criterion $d' \leq R_l$ is satisfied. The synthetically generated samples $\mathbf{b'}_1, \mathbf{b'}_2, \ldots, \mathbf{b'}_m$ corresponding to the clusters $C_1, C_2, \ldots C_m$ of the minority class are then merged together with the initial minority class $\mathscr{S}$ to form the final set $\mathscr{N}$. Henceforth, $\mathscr{N}$ is the new oversampled minority class, and the cardinality of $\mathscr{N}$ is approximately equal to the cardinality of the majority class.

CQNR can also be applied to imbalanced datasets consisting of samples in more than two classes. For b-class classification problem, the class with the maximum number of samples is the majority class while all the other $(b-1)$ classes form the minority classes. CQNR separately processes these $(b-1)$ minority classes for making their cardinalities approximately equal to the cardinality of the majority class. Algorithm 1 describes the working principle of CQNR. As observed, CQNR retains the original dataset and generates the minimum number of synthetic samples required for balancing majority and minority classes, consequently sustaining the distribution of the original dataset. It can handle the oversampling of disjoint clusters of data points of the minority class. It does not eliminate any data point as noise.

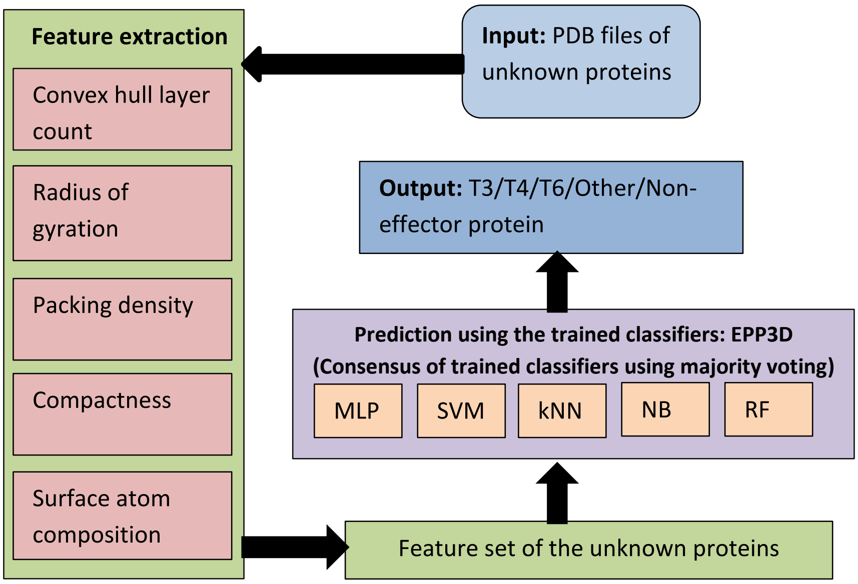
One may argue the biological validity of the synthetic samples. It may be noted that the discovery of effector proteins in several pathogenic species is currently being actively researched, with new effector proteins being discovered frequently. There is no guarantee that these new effector proteins will not resemble the synthetically generated samples. Also, for any sort of class imbalance, even for biological datasets, oversampling has been carried out in multiple investigations of Hu *et al.* [191], Santos *et al.* [342], Zhang *et al.* [450] among others. To ensure maximum resemblance to the experimentally validated data, we have generated synthetic samples in the vicinity of the experimentally verified samples without replicating the samples themselves.

4.2.4 Preprocessing of feature set

Effector proteins in pathogenic bacteria are very less in number, compared to the whole protein set of the pathogen. In such a case, none of the samples can be discarded as noise, since every sample of a dataset may convey useful information. In such a small but potent dataset of effector proteins, undersampling would lead to loss of information. Keeping that in mind,


Feature extraction
Convex hull layer count
Radius of gyration
Packing density
Compactness
Surface atom composition
Input 1: PDB files of experimentally verified effector and non-effector proteins
Training classifiers with oversampled training dataset
MLP
SVM
kNN
NB
RF
Oversampled training dataset using CQNR
Unbalanced training dataset

(a) Training EPP3D


Feature extraction
Convex hull layer count
Radius of gyration
Packing density
Compactness
Surface atom composition
Input: PDB files of unknown proteins
Output: T3/T4/T6/Other/Non-effector protein
Prediction using the trained classifiers: EPP3D
(Consensus of trained classifiers using majority voting)
MLP
SVM
kNN
NB
RF
Feature set of the unknown proteins

(b) Prediction using EPP3D

Figure 4.2: Flowcharts depicting the internal architecture of EPP3D. Figure (a) depicts the stage of training EPP3D. The feature extraction module extracts feature set from PDB files of the experimentally verified effector/non-effector proteins in training phase. After the original dataset is split into a training set and a test set, the training set is balanced by CQNR. This oversampled training dataset is further used to train EPP3D. Figure (b) depicts the prediction phase. The prediction phase starts with the extraction of feature set from PDB files of unknown proteins. The trained EPP3D has been used for prediction of the class label of an unknown protein. The Output module accumulates the outcome of the five classifiers, and determines the class an unknown protein belongs to, based on majority voting.

we have applied algorithm CQNR to balance the effector protein dataset created previously, and utilize the balanced dataset to build EPP3D, the effector protein predictor. Oversampling of the feature set is followed by standardization of the features by subtracting them from the mean followed by scaling them to unit variance. We have used Gini impurity index in a randomized decision tree [401] for feature selection. However, it has been reported that the feature set formed is potent and none of the features are trivial enough to be discarded. Thus all the features have been taken into consideration for development of EPP3D.

4.2.5 Architecture of EPP3D

We have developed a system for classification of various effectors and non-effector proteins, based on their 3D structure. EPP3D is a system that predicts, based on eight features, the class of an unknown protein. The system uses the training dataset, which involves the features extracted from tertiary structure of the experimentally verified effector and non-effector proteins. EPP3D extracts features from the PDB files of the unknown proteins. Due to an imbalanced dataset, it has been oversampled by CQNR. An ensemble of five classifiers, *viz.*, support vector machine (SVM), k-nearest neighbor (k-NN), decision tree (DT), naive-Bayes (NB) and random forest (RF), with majority voting predicts the class of an unknown protein. It is a more suitable alternative in classification over single classifiers [237]. Ensemble learning average out biases, reduce variance and are unlikely to overfit. Hard voting strategy has been used to generate predictions of EPP3D, by taking into consideration the predictions of individual classifiers. As an output, it predicts whether a protein belongs to class 1 (T3), class 2 (T4), class 3 (T6), class 4 (T1, T2, T7) or class 5 (non-effectors). Cross-validation has been implemented for parameter tuning of the individual classifiers. The flow of EPP3D has been depicted in Figure 4.2.

4.3 Results

The effectiveness of CQNR has been demonstrated on some of the profoundly referenced benchmark datasets, namely, Pima Indians Diabetes, Haberman, Spambase, Hill-Valley, and Blood Transfusion datasets as given in Table 4.4. These datasets have been downloaded from UCI machine learning repository [120]. Besides, we have also generated three highly imbalanced synthetic datasets. The superior performance of CQNR has been exhibited on the datasets mentioned above over some existing oversampling algorithms, namely, random oversampling, SMOTE, Borderline-SMOTE, C-SMOTE, and Safe-level-SMOTE. We have also worked on various effector protein datasets. For this purpose, we have, first of all, extracted and analyzed features from various experimentally verified effector proteins. The

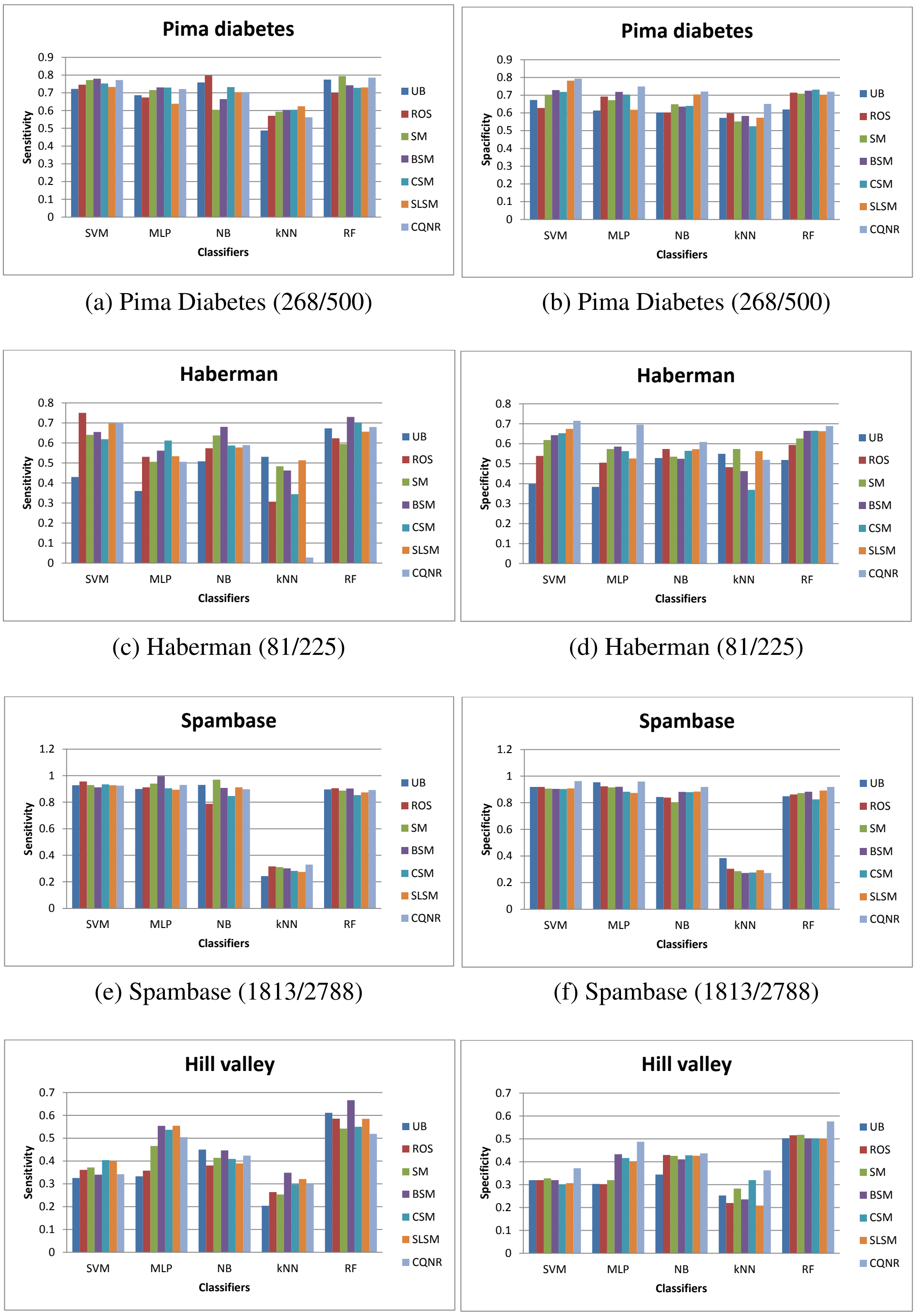
Table 4.4: Summary of the imbalanced datasets (2-class) used to compare the performance of various oversampling techniques.

Dataset	Number of Features	Majority Class cardinality	Minority Class cardinality
Pima Diabetes	8	500	268
Haberman	3	225	81
Spambase	57	2788	1813
Hill-Valley	100	612	329
Blood transfusion	4	570	178
Synthetic Dataset 1	2	400	100
Synthetic Dataset 2	2	500	100
Synthetic Dataset 3	2	600	100

datasets have been balanced by CQNR. Finally, the effector proteins have been classified using five popular classification algorithms.

4.3.1 Application of CQNR for balancing various benchmark datasets along with comparison

The performance of CQNR has been demonstrated on the five benchmark datasets, namely, Pima Diabetes, Haberman, Spambase, Hill-valley, and Blood Transfusion. Besides, we have designed three synthetic datasets which are highly imbalanced. To assess the performance of the oversampling methods, we have considered five classification algorithms, namely, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), Naive Bayes (NB) classifier, k-Nearest Neighbor (kNN) classifier, and Random Forest (RF) classifier. For SVM, the decision function used is one-over-one ('ovo') with RBF kernel. For MLP, we have considered two hidden layers, apart from the input and output layers. For kNN, we have considered k=3. We have first split the entire dataset into two sets - training set and test set. The training set comprises 70% of the entire dataset while remaining 30% samples form the test set. We have kept the test set (30%) aside for testing purpose. The training set (70%) has been oversampled. A 10-fold cross validation has been carried out on the oversampled training set. That is, the oversampled training set has been divided into ten non-overlapping subsets. The classifiers have then been trained using nine such subsets while the remaining subset has been used for validation. When considering the three highly unbalanced synthetic datasets, the accuracy is maximum for classification of unbalanced dataset. As explained in Chapter 1, accuracy is not sensitive to imbalanced data [391]. We have tabulated $Specificity$, $Sensitivity$, F-score and G-mean of CQNR against the other popularly used oversampling



(a) Pima Diabetes (268/500)

(b) Pima Diabetes (268/500)

(c) Haberman (81/225)

(d) Haberman (81/225)

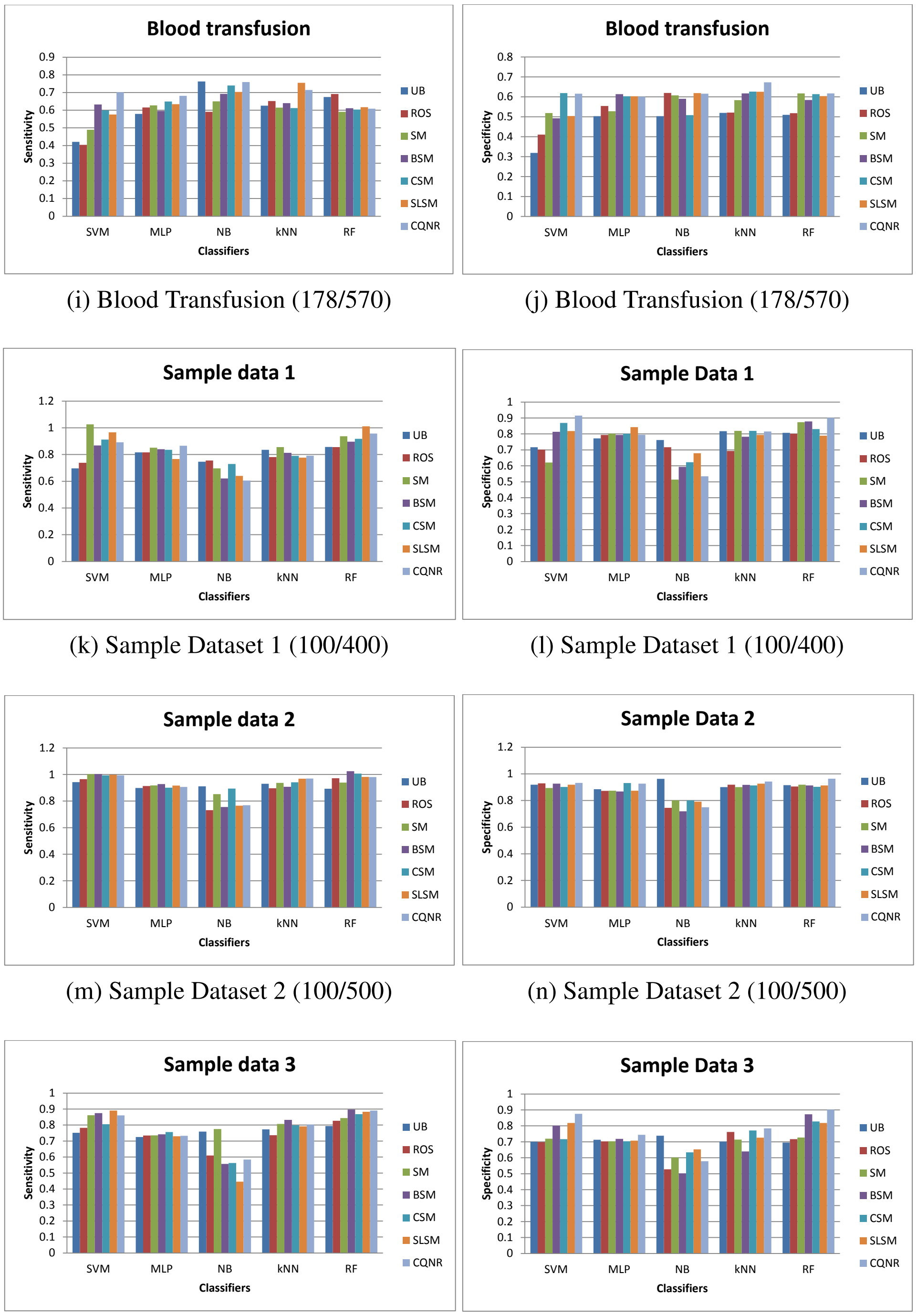
(e) Spambase (1813/2788)

(f) Spambase (1813/2788)

(g) Hill-valley (329/612)

(h) Hill-valley (329/612)

Figure 4.3: Comparison of the classification performances, in terms of *Sensitivity* and *Specificity*, of different classifiers on datasets oversampled by different oversampling algorithms. The numbers within brackets indicate the ratio of the cardinalities of minority class to the majority class. The abbreviations for the methods are UB - imbalanced, ROS - Random Oversampling, SM - SMOTE, BSM - borderline SMOTE, CSM - C Smote, SLSM - Safe level SMOTE, CQNR - Cluster Quality-based Non-Reductional Oversampling. As observed from the figures, CQNR has performed the best over the other oversampling algorithms considered here.



(i) Blood Transfusion (178/570)

(j) Blood Transfusion (178/570)

(k) Sample Dataset 1 (100/400)

(l) Sample Dataset 1 (100/400)

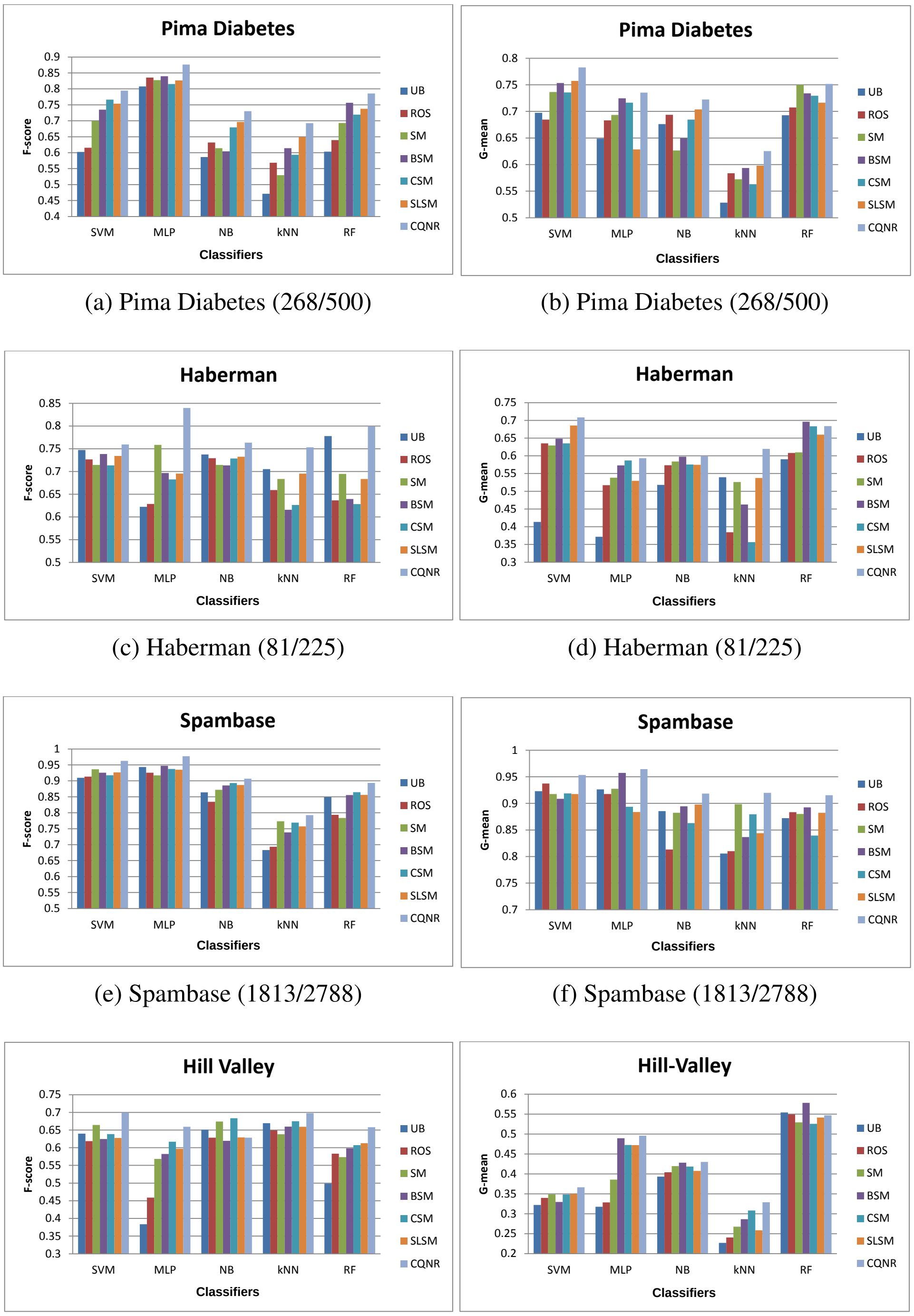
(m) Sample Dataset 2 (100/500)

(n) Sample Dataset 2 (100/500)

(o) Sample Dataset 3 (100/600)

(p) Sample Dataset 3 (100/600)

Figure 4.3: Comparison of the classification performances, in terms of *Sensitivity* and *Specificity*, of different classifiers on datasets oversampled by different oversampling algorithms. The numbers within brackets indicate the ratio of the cardinalities of minority class to the majority class. The abbreviations for the methods are UB - imbalanced, ROS - Random Oversampling, SM - SMOTE, BSM - borderline SMOTE, CSM - C Smote, SLSM - Safe level SMOTE, CQNR - Cluster Quality-based Non-Reductional Oversampling. As observed from the figures, CQNR has performed the best over the other oversampling algorithms considered here.



(a) Pima Diabetes (268/500)

(b) Pima Diabetes (268/500)

(c) Haberman (81/225)

(d) Haberman (81/225)

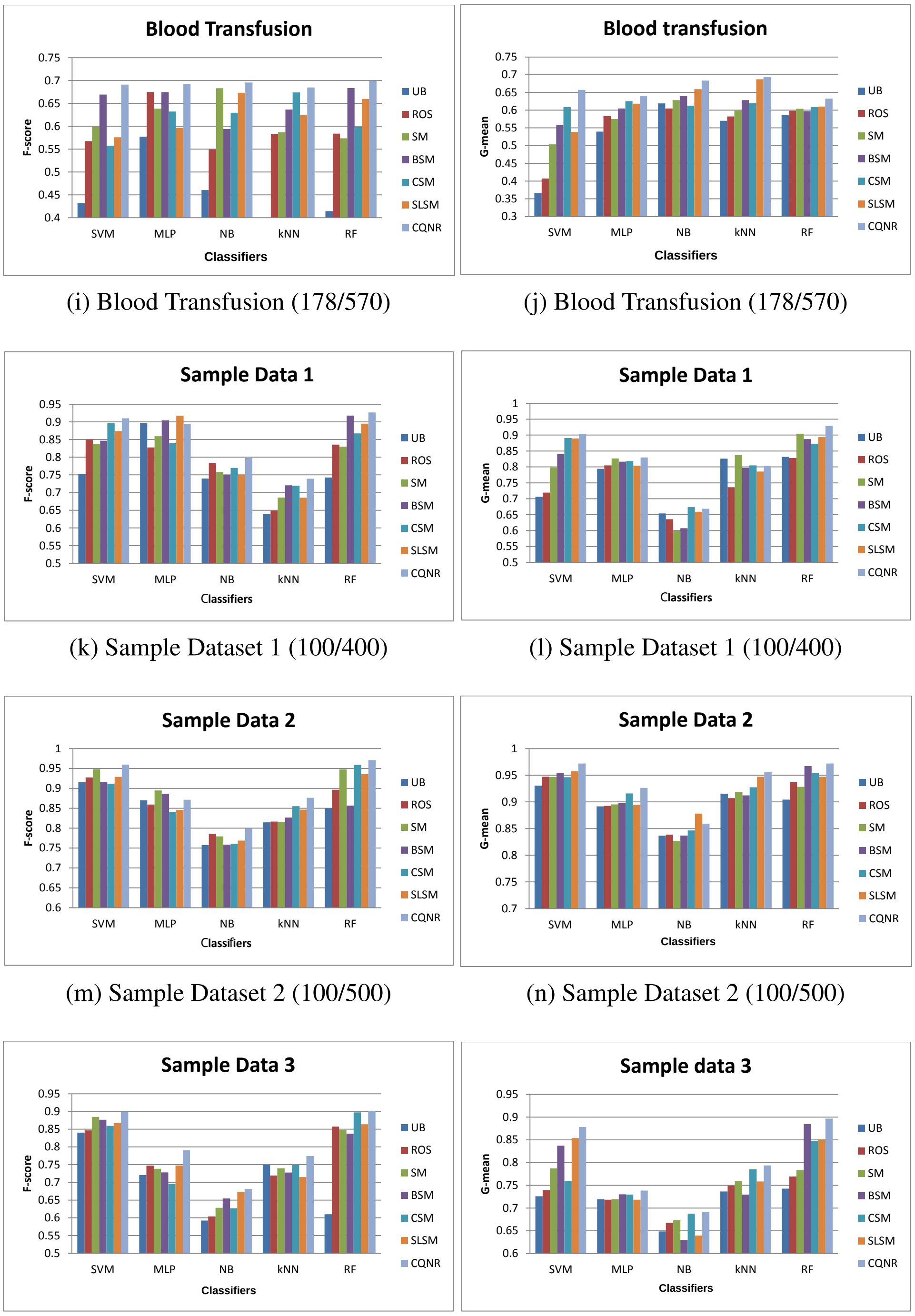
(e) Spambase (1813/2788)

(f) Spambase (1813/2788)

(g) Hill-valley (329/612)

(h) Hill-valley (329/612)

Figure 4.4: Comparison of the classification performances, in terms of F-score and G-mean, of different classifiers on datasets oversampled by different oversampling algorithms. The numbers within brackets indicate the ratio of the cardinalities of minority class to the majority class. The abbreviations for the methods are UB - imbalanced, ROS - Random Oversampling, SM - SMOTE, BSM - borderline SMOTE, CSM - C Smote, SLSM - Safe level SMOTE, CQNR - Cluster Quality-based Non-Reductional Oversampling. As observed from the figures, CQNR has performed the best over the other oversampling algorithms considered here.



(i) Blood Transfusion (178/570)

(j) Blood Transfusion (178/570)

(k) Sample Dataset 1 (100/400)

(l) Sample Dataset 1 (100/400)

(m) Sample Dataset 2 (100/500)

(n) Sample Dataset 2 (100/500)

(o) Sample Dataset 3 (100/600)

(p) Sample Dataset 3 (100/600)

Figure 4.4: Comparison of the classification performances, in terms of F-score and G-mean, of different classifiers on datasets oversampled by different oversampling algorithms. The numbers within brackets indicate the ratio of the cardinalities of minority class to the majority class. The abbreviations for the methods are UB - imbalanced, ROS - Random Oversampling, SM - SMOTE, BSM - borderline SMOTE, CSM - C Smote, SLSM - Safe level SMOTE, CQNR - Cluster Quality-based Non-Reductional Oversampling. As observed from the figures, CQNR has performed the best over the other oversampling algorithms considered here.

algorithms, depicted in the Figures 4.3 and 4.4.

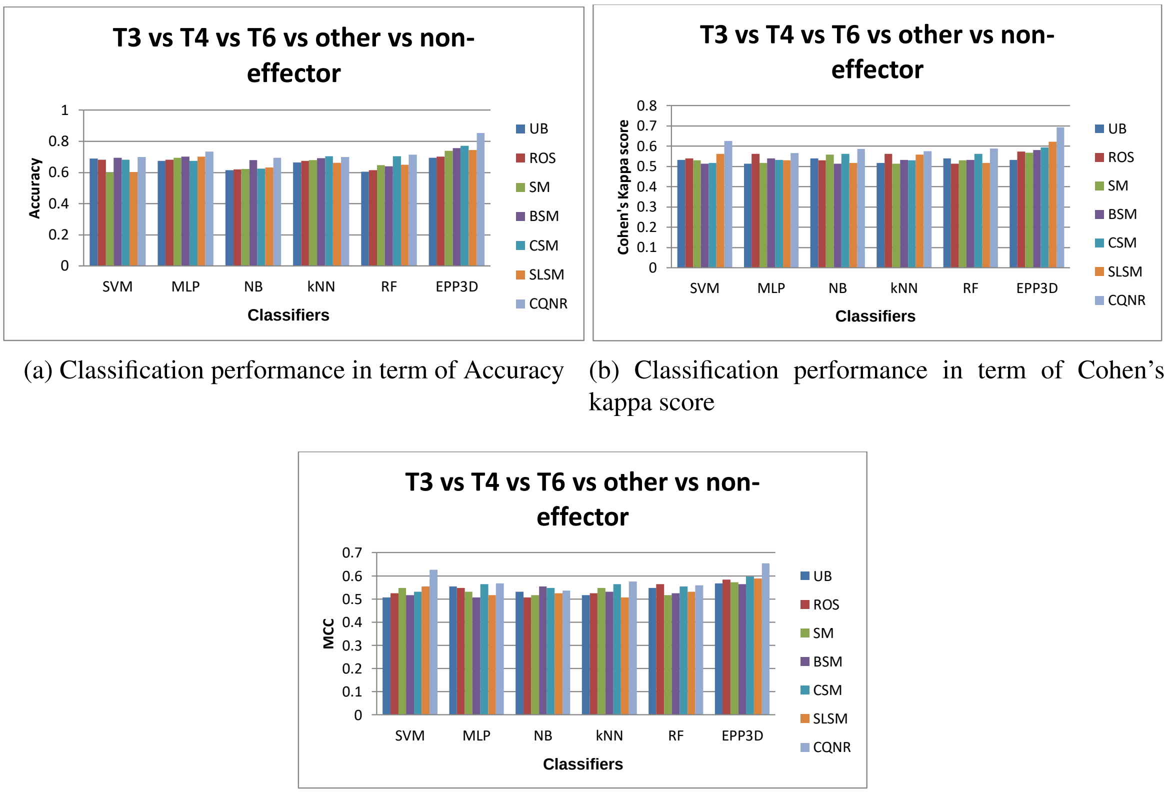
As observed from the plots in Figure 4.4, CQNR has achieved the highest F-score of 0.84 for Pima Diabetes dataset, 0.95 for Spambase, 0.69 for Blood Transfusion datasets, 0.92 for Synthetic Dataset 1 and 0.97 for Synthetic Dataset 2. CQNR has achieved the highest G-mean score of 0.78 for Pima Diabetes dataset, 0.71 for Haberman dataset, 0.69 for Blood Transfusion dataset, 0.92 for Synthetic Dataset 1, 0.97 for Synthetic Dataset 2 and 0.89 for Synthetic Dataset 3. CQNR has outperformed, in terms of F-score and G-mean, the other oversampling algorithms for almost all the datasets using all the five classification algorithms. The performance of numerous oversampling algorithms, including that of CQNR on five benchmark datasets and three synthetic datasets, has been given in Tables A.3 to A.7 in Appendix A.

CQNR has led to more reliable performance for different datasets and classification techniques, with some exceptions (Figure 4.4). For Haberman and Hill-valley datasets, borderline-SMOTE has shown the best performance among all the other oversampling algorithms for random forest classifier with respect to G-mean. For almost all the datasets, CQNR shows a stable performance in terms of F-score and G-mean, indicating an unbiased prediction of samples. For Pima dataset, the variation of F-score is low, which suggests that nearly all the oversampling algorithms have a negligible difference in performance. On the other hand, Blood transfusion dataset has shown a drastic difference for performance metric F-score for various oversampling algorithms. Hill-valley dataset, in terms of G-mean, has shown a stable performance over all the oversampling algorithms with a negligible difference. For Haberman dataset, on the other hand, the oversampling algorithms have resulted in a drastic difference in performance.

4.3.2 Comparative performance of EPP3D on various effector protein datasets balanced by some existing oversampling methods including CQNR

Several classification algorithms have been used to classify the experimentally verified T3, T4, and T6 effector proteins against other effector and non-effector proteins. The dataset consists of 36 T3 effector proteins, 80 T4 effector proteins, 31 T6 effector proteins, 24 effectors of T1, T2, T5 and T7, and 120 non-effector proteins from the non-pathogenic bacteria *Bacteriodes vulgatus* and *Listeria innocua*. Thus, the dataset is visibly imbalanced. In such a small but potent dataset of effector proteins, undersampling would lead to loss of information. Keeping this fact in mind, we have applied algorithm CQNR to balance the dataset by oversampling.

After balancing, the feature set has been normalized. We have subjected the dataset to



(a) Classification performance in term of Accuracy

(b) Classification performance in term of Cohen's kappa score

(c) Classification performance in term of MCC

Figure 4.5: Performance comparison of various classification algorithms on effector and non-effector proteins, after balancing the dataset by different oversampling methods. The abbreviations for the methods are UB - imbalanced, ROS - Random Oversampling, SM - SMOTE, BSM - borderline SMOTE, CSM - C Smote, SLSM - Safe level SMOTE, CQNR - Cluster Quality-based Non-Reductional Oversampling. As observed from the graphs, CQNR with consensus-based classifier (EPP3D) has provided superior performance over the other oversampling algorithms while classifying the effectors.

variance threshold, which removes features with low variance. None of the features in the effector protein dataset has ultimately been removed. We have reported accuracy, MCC, and κ score of the five commonly used techniques along with EPP3D for classification of T3, T4, and T6 effector proteins. These five techniques are Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), Naive Bayes (NV), K-Nearest Neighbor (KNN) and Random Forest (RF). It has been found that EPP3D has resulted in a remarkable improvement in the performance of predicting unknown proteins into different classes.

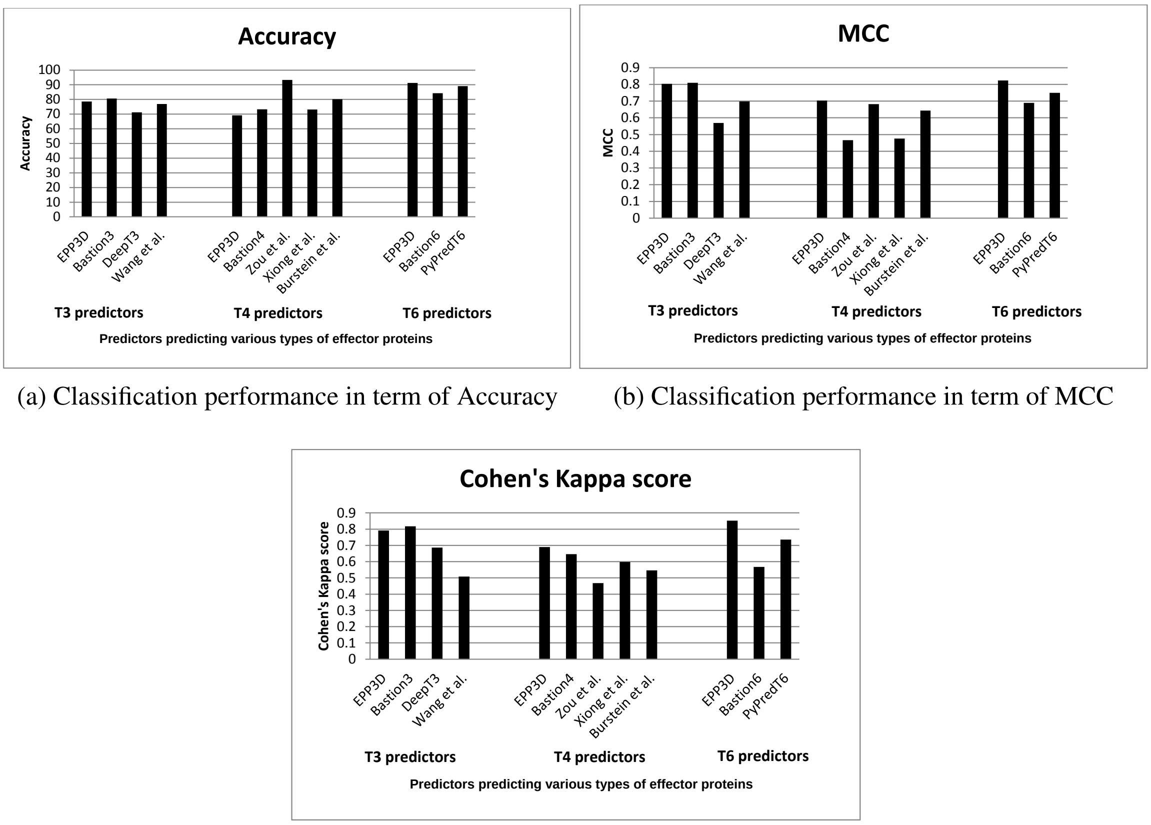
In Figure 4.5, we have assessed the performance of EPP3D, in terms of accuracy, κ score and MCC, to know how diverse the 3D structural characteristics of the pathogenic effectors pertaining to different secretion systems are. Classification of effectors by EPP3D, where CQNR has balanced the effector dataset, has resulted in the best performance in terms of accuracy (85.43%), MCC (0.6536), and κ score (0.6821). κ score ranging from 0.61 to

0.80 indicates a substantially well performing predictor [144]. As observed from Figure 4.5, CQNR has resulted in performances having an accuracy ranging from 69.94% for SVM to 85.43% for EPP3D, while the imbalanced dataset has produced an approximate performance ranging from 68.97% for SVM to 69.43% for EPP3D. CQNR has led to an average classification performance of 73.29%, the highest among all the other oversampling techniques. EPP3D, along with CQNR, has obtained an average accuracy of 75.18%, the highest among the performances of individual classifiers and oversampling algorithms. However, naive-Bayes classifier has given better performance for borderline-SMOTE compared to the other oversampling algorithms. A visible improvement has been noticed in classification accuracy using balanced data compared to that using imbalanced data for most of the classifiers. The performance of EPP3D with CQNR has provided an overall better accuracy, MCC value, and κ score than that of individual classifiers.

Several subsets of proteins oversampled by CQNR have been classified using EPP3D. A detailed tabulated representation of the performance of various oversampling algorithms has been given in Table A.8 to A.10 of Appendix A. As observed from the tables, the dataset, where any one of T3, T4 and T6 has been considered as a single class versus the “Others”, shows the best classification performance with respect to accuracy (96.24%), MCC (0.8834) and κ score (0.8624). Similar values across all these three measurements indicate a stable performance of CQNR. On the other hand, the subsets of the 3-class dataset (T4, T3, and T6) combined show an unsatisfactory performance, in terms of accuracy (64.32%), MCC (0.5432) and κ score (0.5932). Such a poor performance indicates that the features of the 3-class dataset (T3, T4, and T6) have considerably low variance. CQNR, together with EPP3D, has shown better performance for majority of the effector protein datasets.

4.3.3 Comparative performance of EPP3D with existing effector protein prediction algorithms

Several methods have been developed to classify effector proteins based on their peptide sequences. These include ones using machine learning techniques [59, 353, 406, 408, 409, 432, 433, 440, 456]. So far, no work has been reported, which predicts the classes of effector proteins based on 3D structural characteristics of experimentally verified effectors. Absence of a 3D structure-based effector protein predictor has led to the designing of EPP3D. The primary data and the feature set of the aforesaid existing methods are completely different from that used in EPP3D. In order to compare the performance of these existing methods with EPP3D, we have collected the peptide sequences of the corresponding PDB structures of T3, T4 and T6 effectors. A summary of the comparison has been depicted in Figure 4.6. A detailed tabulated representation of the same has been provided in Table 4.5.



(a) Classification performance in term of Accuracy

(b) Classification performance in term of MCC

(c) Classification performance in term of Cohen's Kappa (κ) score

Figure 4.6: Comparison of classification performance of EPP3D with different effector predictors.

Classification of T3 effector proteins using EPP3D has resulted in an accuracy of 78.65%, MCC of 0.8026 and κ score of 0.7913; whereas Bastion3 has resulted in an accuracy of 92.7%, MCC of 0.809 and κ score of 0.8174. DeepT3 has resulted in an accuracy of 81.2%, MCC of 0.569 and κ score of 0.6864. The technique developed by Wang *et al.* has obtained an accuracy of 86.88%, MCC of 0.6979 and κ score of 0.5079. Here Bastion3 has shown the highest accuracy, MCC value and κ score.

EPP3D has classified T4 effector and non-effector proteins with an accuracy of 69.24%, MCC of 0.7038 and κ score of 0.6893. The algorithm developed by Burstein *et al.* [59] has achieved an accuracy of 80.2%, MCC of 0.643 and κ score of 0.546 for prediction of T4 effectors, while the method of Zou *et al.* [456] has reported an accuracy of 93.3%, MCC of 0.682 and κ score of 0.4679. Bastion4 predictor has rendered an accuracy of 73.3%, a low MCC of 0.466 and κ score of 0.6457. The method of Xiong *et al.* has resulted in an accuracy of 73.2%, MCC of 0.476 and κ score of 0.5975. Here, EPP3D has shown the best performance with respect to MCC and κ score, while Zou *et al.* has shown the best performance with respect to accuracy.

Table 4.5: Performance comparison of T3, T4 and T6 effector protein predictors

Effector type	Predictor	Accuracy	MCC	κ score
T3	EPP3D	78.65	0.8026	0.7913
	Bastion3	80.7	0.809	0.8174
	DeepT3	71.2	0.569	0.6864
	Wang *et al.*	76.88	0.6979	0.5079
T4	EPP3D	69.24	0.7034	0.6893
	Bastion4	73.3	0.466	0.6457
	Zou *et al.*	93.3	0.782	0.4679
	Xiong *et al.*	73.2	0.476	0.5975
	Burstein *et al.*	80.2	0.643	0.5467
T6	EPP3D	91.23	0.8233	0.8523
	Bastion6	84.3	0.689	0.568
	PyPredT6	89.12	0.7492	0.736

T6 effector proteins have been classified by EPP3D with an accuracy of 91.23%, MCC of 0.8233 and κ score of 0.8523. Bastion6 predictor has provided an accuracy of 84.3%, MCC of 0.689 and κ score of 0.568. PyPredT6 has reported an accuracy of 89.12%, MCC of 0.7492, and κ score of 0.736. EPP3D has provided much better accuracy in classifying T6 effector proteins based on their 3D structures.

In order to assess and compare the performance of the aforesaid existing methods with EPP3D, we have considered three individual lists for each of T3, T4 and T6 effector proteins. Each list contains 20 independent non-overlapping experimentally verified effectors proteins.

Among 20 T3 effector proteins, EPP3D has been able to predict 15 proteins correctly, whereas Bastion3 [406] has been able to predict 17 proteins. DeepT3 [433] has predicted 12 proteins correctly. The method of Wang *et al.* [412] has been able to predict 11 T3 proteins. Among 20 T4 effector proteins, EPP3D has been able to predict 18 proteins correctly. Bastion4 [408], however, was unable to generate any predictive result. The algorithm developed by Zou *et al.* [456] has been able to predict 12 proteins correctly, while Xiong *et al.* [432] have predicted 13 proteins correctly, and Burstein *et al.* [59] have predicted 11 out of 20 proteins correctly. Among 20 T6 effector proteins, EPP3D has been able to predict 17 proteins correctly. Bastion6, however, has been unable to generate any predictive result. PyPredT6 [353] has been able to predict 14 T6 effector proteins correctly. The summary of the results has been provided in Tables A.11 to A.13 in Appendix A.

4.4 Discussion

In this section, we discuss the qualitative comparison of CQNR and EPP3D with the current state-of-the-art investigations.

4.4.1 Comparison of CQNR with other oversampling algorithms

CQNR has been compared with five existing oversampling methods, namely, Random oversampling [232], SMOTE [77], Borderline-SMOTE [175], C-SMOTE [181], and Safe-level-SMOTE [57]. In the random oversampling algorithm, minority class samples are duplicated at random, such that the majority and the minority classes become balanced, thereby leading to a severe drawback of overfitting.

Generation of overfitted classifiers due to random oversampling has led to the development of SMOTE [77]. In SMOTE, an entirely new synthetic dataset is conceived from the original minority dataset to form a new set containing the original samples and new synthetic samples. However, a high SMOTE rate may lead to overfitting and adversely influence the prediction performance of the minority class. SMOTE has randomly generated synthetic points, and many of them have been generated in the region where minority class samples do not exist.

Borderline-SMOTE [175], another oversampling method, is a tweak of SMOTE, designed to do away with the ambiguities of SMOTE. Borderline-SMOTE is exclusively applicable to datasets, where the number of borderline samples is low. Borderline-SMOTE [175] has divided the points into three categories - noise, danger, and safe. Noise samples of a minority class are those that have a maximum number of majority class samples as their nearest neighbors. Danger samples are the borderline samples having a mixture of minority and majority class samples as their nearest neighbors. Safe samples have the maximum number of minority class samples as their nearest neighbors. It oversamples only the borderline samples of the minority class. A limitation of borderline-SMOTE is the following. If the number of danger samples is low compared to the others, the synthetic samples generated by borderline-SMOTE may not balance the final dataset. In such a scenario, the number of danger samples will have to be large enough, which would lead to clustering of synthetic data around the limited boundary samples.

C-SMOTE [181] comprises the same procedure as SMOTE [77], except that it has generated the best SMOTE rate such that the classification results in maximum accuracy. The method uses a classifier ensemble to attain an optimal SMOTE rate and implement oversampling based on this SMOTE rate. For the present datasets, the number of synthetic samples generated is more than the number of majority class samples.

Safe-Level-SMOTE [57], another variation of SMOTE, splits the initial minority class samples into three categories, safe, borderline, and noise, and discards the noise samples. Only the safe synthetic samples have been considered for oversampling. Safe-Level-SMOTE generates a synthetic sample in the space densely populated by the original samples, which may lead to overfitting. Hence, the generation of synthetic samples has been restricted to the center of the dataset.

A significant drawback of all the above algorithms is that if a minority class is clustered, these algorithms may generate synthetic samples between these clusters. None of these algorithms ensure generation of synthetic samples in the vicinity of the minority class samples and not near majority class samples. The area outside the clusters of minority class samples may belong to the majority class. CQNR checks whether the distance between the cluster center and a new synthetic sample generated in that particular cluster is less than the radius of the cluster. If yes, the synthetic sample is added to the minority class; if no, the sample is discarded, and a new sample is generated. Another major drawback of some of these algorithms is that they eliminate samples which are noise. In biological datasets, deletion of any samples as noise would lead to loss of crucial information. CQNR does not eliminate any samples as noise, thus keeping the original dataset intact.

For different oversampling algorithms, reasonable sets of parameter values have been experimentally determined and used. For borderline-SMOTE [175] and safe-level-SMOTE [57], $k = 5$, and the number of random samples to be selected from k-neighbors has been taken as three. The threshold value used by these algorithms, for deciding whether a minority class sample is noise, danger, or safe has been set to six. In other words, if the number of majority class samples in k-nearest neighbors of a minority sample is less than the threshold value, the sample is said to be safe. If the number of majority class samples in k-nearest neighbors of a minority sample is equal to the threshold value, the samples are said to be borderline. On the other hand, if the number of majority class samples in k-nearest neighbors of a minority sample is more than the threshold value, the minority sample is classified as noise. The number of clusters, predefined in C-SMOTE [181], has been set to six.

4.4.2 Comparison of EPP3D with other effector protein predictors

As mentioned in Section 3.4, a few attempts have been made towards classification of effector proteins based on their peptide sequences [59,353,406,408,409,432,433,440,456]. Prediction of T3 effector proteins in genomes of gram-negative bacteria has been done by Yang *et al.* The authors have used Support Vector Machine (SVM) on N-terminal of amino acid sequences to predict novel T3 effector proteins [440]. A two-layered ensemble predictor, called Bastion3 [406], has predicted T3 effector proteins. Bastion3 is based on the features obtained from N-terminal of the proteins. Another investigation of Wang *et al.* [412] has used SVM to predict effector proteins based on the features obtained from N and C-terminals of the proteins. Xue *et al.* [433] have used deep learning framework, called DeepT3, to predict T3 effector proteins taking only the first 100 residues for prediction. Bastion3 has shown the maximum *accuracy, MCC*, and κ score.

Identification of T4 effector proteins has been made based on amino acid composition. Zou *et al.* [456] have used SVM to predict T4 effector proteins. In the investigation of Burstein *et al.*, the ORFs of the proteins in *Legionella pneumophila* have been classified either as effector or non-effector proteins using a machine learning approach [59]. Xiong *et al.* [432] and Wang *et al.* [408] have predicted T4 effectors using ensemble classifiers based only on C-terminal features. The latter group has developed Bastion4 to predict T4 effectors [408]. EPP3D has shown the best performance with respect to *MCC* and κ score, while Zou *et al.* has shown the best performance with respect to *accuracy*. For identification of T6 effector proteins, Bastion6, an SVM-based T6 effector protein predictor [409], and PyPredT6 [353], an ensemble learning-based predictor [353] are the two currently available tools. EPP3D has provided much better prediction accuracy for T6 effector proteins.

EPP3D, based on their 3D structural features, has reported stable performance in terms of the performance measures. However, such a trend is not noticed for the other classifiers, except for Bastion3, the classifier that classifies T3 effectors and non-effectors. Another issue has been noticed regarding consideration of the non-effector dataset. For example, Bastion6 has considered the non-effector set of Zou *et al.* The method of Zou *et al.* classifies T4 effectors and non-effectors, where the non-effectors are those that are not T4 effectors. Due to the multi-functional nature of prokaryotic genes [206], this may not be a reliable approach. Proteins that are not T4 effectors may have an association with T6SS machinery. Likewise, Yang *et al.* have extracted the effectors from *P. syringae*, and the remaining proteins from the entire genome have been treated as non-effectors. In contrast to these, we have taken the non-effector dataset from an experimentally verified non-pathogenic organism.

4.5 Conclusions

In this chapter, we have developed a novel oversampling technique, called CQNR, and an effector protein predictor, called EPP3D, based on 3D structure of effector proteins. We have depicted how the application of the oversampling technique has helped in better classification of the effectors and the development of EPP3D. The experiments show that CQNR effectively has resolved the shortcomings of some existing algorithms. CQNR has resulted in superior performance over some existing algorithms as well as sustained the essence of the original dataset.

In order to demonstrate the effectiveness of the present method, we have considered a dataset derived from 3D structures of experimentally verified T3, T4, and T6 effector proteins. Only 3D structural patterns have been considered here as earlier investigations have already reported the inconclusive nature of 1D (amino acid sequence) and 2D (alpha helices, beta sheets, and random coils among others) features in differentiating T3 and T4 effectors. These feature patterns can be used for distinguishing the known effector proteins from non-effectors as well as discovering the novel effector proteins. The sample size here is considerably limited since only a few known resources are available for effector proteins.

We have also developed EPP3D for classification of unknown proteins into T3, T4, and T6 effector proteins against other secreted proteins (T1, T2, T5, T7) and non-effector proteins. Since the original training dataset is imbalanced, we have used CQNR to balance the dataset. A considerable improvement in classification performance of effector proteins after applying CQNR and a consensus of classifiers, has been reported. As a future scope, we intend to incorporate more features based on 3D structure of effector proteins along with the existing ones to develop a more robust classifier. Discovery of new secretion systems will instigate the discovery of effectors. These newly discovered effectors can be included in future to design a more versatile classifier.

Effectors in gram-positive bacteria are primarily secreted by T7 secretion systems [1, 416]. In literature, we were unable to find any algorithm for prediction of T7 effector proteins. Hence, in the next chapter, we have come up with a deep neural network-based system, called DeepT7, to uniquely identify T7 effector proteins, based on their primary and secondary structures.

Chapter 5

DeepT7: A Deep Neural Network System for Identification of Type VII Effector Proteins [351]

5.1 Introduction

In Chapter 4, an algorithm to identify effector proteins based on their 3D structure has been developed. Due to unavailability of substantial information pertaining to the 3D structure of experimentally verified Type VII (T7) effector proteins, it could not be considered as an individual class in the identification of types of effectors. T7 effectors, secreted by gram-positive bacteria [1, 416], are highly pathogenic in nature. Examples of T7 effector proteins are the proteins secreted by *Mycobacterium tuberculosis*, which are known to cause the infectious disease Tuberculosis. Additionally, we have noticed the absence of any *in silico* techniques to uniquely identify T7 effectors in literature. As we were unable to find 3D structure-based information and any T7 effector identification techniques, we aim to develop, in this chapter, a system to identify T7 effectors based on their primary and secondary structure information.

The existence of T7SS was discovered in 2007 [1, 416] in gram-positive bacteria. T7SS has been known to exist in *M. tuberculosis*, *M. bovis*, *Streptomyces coelicolor* and *S. aureus*. T7 systems show a more restricted distribution, and are typical for mycobacteria and other high GC-content[1] *Actinobacteria*. Effects of reported T7SS effector proteins include the suppression of pro-inflammatory responses by modulating macrophage response [378], necrosis [203], apoptosis [111], membrano-lysis [104], cytolysis [164, 190], and preventing the restriction of intracellular bacterial growth by the host [278].

In gram-negative bacteria, bacterial effector proteins are primarily secreted by T3SS,

[1] GC-content (or guanine-cytosine content) is the percentage of guanine or cytosine in a DNA or RNA molecule.

T4SS, and T6SS [14]. In gram-positive bacteria, effectors are primarily secreted by T7 secretion systems [1, 416]. *In silico* identification of effector proteins of T3SS, T4SS and T6SS have been extensively investigated [354]. However, no *in silico* method has been reported so far for identification of T7 effector proteins to date.

Investigations involving computational prediction of T3 effector proteins has been done previously [17, 261, 338, 407, 433, 440]. Identification of secretion signals in T3 effector proteins of gram-negative bacteria using an Artificial Neural Network (ANN) with gradient descent back-propagation learning with momentum and Support Vector Machine (SVM), has been done by Löwer *et al.* [261]. Samudrala *et al.* [338] have used SVM to predict the T3 effectors. Yang *et al.* have applied SVM on features extracted from the N-terminal of T3 effector proteins for their identification [440]. Identification of T3 secreted proteins has been implemented by Arnold *et al.* [17]. Wang *et al.* [407] have developed a two-layered ensemble T3 effector protein predictor Bastion3, based on the features obtained from N-terminal of T3 effector proteins. Another investigation by Xue *et al.* [433] have used the first 100 residues of T3 effector proteins, in deep learning framework, to predict T3 effector proteins.

Multiple investigations regarding the identification of T4 effectors have been carried out simultaneously alongside the identification of T3 effectors. Zou *et al.* [456] have reported an SVM-based method to identify T4 effector proteins based on amino acid composition. In *L. pneumophila*, the ORFs of proteins have been used by Burstein *et al.* to identify T4 effector proteins by using a machine learning approach [59]. They have used genomic, evolutionary, regulatory networks and pathogenic features from ORFs, to identify T4 effector proteins. Xiong *et al.* [432] and Wang *et al.* [413] have predicted T4 effectors using ensemble classifiers exclusively based on C-terminal features. This investigation has led to the development of a T4 predictor called Bastion4 [413].

Bastion6, an SVM-based protein predictor [409], and PyPredT6, an ensemble learning-based effector protein predictor [353], are currently the two available tools for the identification of T6 effector proteins. However, Bastion6 reports several limitations. These limitations pertain to its choice of non-effector proteins, input size, functionality of the server, choice of classifier, reliability of the results, speed of execution, its predicted effectors among others.

There are four major points of concern with many of the techniques cited above. The set of proteins considered to train the classifiers for identification of effectors consists of hypothetical proteins. The functionality of such hypothetical proteins has not yet been discovered. Since we cannot be sure if they are effectors or not, including them either in the training or test set is not entirely justified [353]. Second, proteins that were known to be effectors of a different type have been included as non-effectors while predicting effectors of a particular type. For example, investigations that have attempted to classify T3/T4 effectors have included proteins secreted by T1SS, T2SS, T5SS, T6SS, and T7SS as non-effectors.

However, the multi-functional nature of proteins indicates that their inclusion in the dataset may not be a good decision [314]. Third, not all methods applied feature selection techniques during their experimentation, and this may lead to the possibility of overfitting [355]. Fourth, the corresponding nucleotide sequences of effector proteins have not been taken into consideration in any of the investigations above, except in PyPredT6. Nucleotide sequences hold crucial information regarding the functionality of their corresponding proteins [53].

In this chapter, we introduce a deep neural network framework, called DeepT7, for identification of T7 effector proteins by utilizing a set of 1727 features. The chapter begins with data collection, followed by feature extraction. We have extracted features from nucleotide and amino acid sequences of experimentally verified T7 effector proteins obtained from three prominent databases, namely, KEGG [209], UniProt [90] and NCBI [92]. The feature set has then been subjected to multiple preprocessing steps in order to build a robust identification system. We further go on to describe the architecture of DeepT7. DeepT7 has been regularized and extensively cross-validated to prevent overfitting; thereby resulting in a highly reliable T7 effector protein identification system. The efficiency and credibility of the system have been endorsed by the biological validation of the effectors identified by DeepT7 from two gram-positive pathogenic bacteria, namely, *Mycobacterium bovis* and *Streptococcus pneumoniae*.

5.2 Methodology

In this section, we focus on the design of DeepT7. DeepT7 is a standalone system that reads nucleotide and amino acid sequences in FASTA format (Appendix B.1) as input. In the data collection phase, amino acid and nucleotide sequences of T7 effectors and non-effectors have been accumulated. To develop a potent feature set for identification of T7 effectors, the set of accumulated sequences are then further filtered to eliminate hypothetical and putative proteins.

The next step towards the development of DeepT7 is feature extraction. The properties of proteins from which features have been extracted are the composition of amino acids, order of amino acids, secondary structure-based information, solvent accessibility-based information, physicochemical properties and evolutionary information. The feature set formed with these features is imbalanced, due to which it is rectified by the oversampling algorithm, called Cluster Quality-based Non-Reductional (CQNR) oversampling technique, developed in the previous chapter. To prevent overfitting and improve the performance of DeepT7, feature selection has been performed on the extensive feature set based on Gini impurity index.

A deep neural network-based system has been chosen for design of DeepT7. An ensemble model consisting of support vector machine (SVM), k-nearest neighbor (k-NN), decision tree (DT), naive-Bayes (NB) and random forest (RF) and deep neural network (dNN), has been taken into consideration. However, due to its poor performance compared to the deep neural network, it has not been chosen as the model for T7 effector identification. Cross-validation has been implemented for parameter tuning of the deep neural network. DeepT7 has undergone extensive training and testing for accurate identification of T7 effector proteins. DeepT7 is a standalone application, which can be downloaded from the website http://projectphd.droppages.com/DeepT7.html/.

5.2.1 Data collection

A set of 209 experimentally verified T7 effector proteins has been accumulated from KEGG [209], Uniprot [90] and NCBI [92]. The set of T7 effector proteins used in this study comprises those obtained from multiple species of gram-positive bacteria. On the other hand, the non-effector set is the complete proteome of *Faecalibacterium prausnitzii*, which is a non-pathogenic gram-positive bacteria that live in human gut [183]. Most of the effector classification investigations use proteins of the same pathogenic species, with housekeeping proteins as the non-effector class, and effector proteins as the effector class. However, prokaryotic genes are mostly multi-functional [206,314], and thus there may be a possibility that the housekeeping proteins exhibit some effector characteristics. In order to avoid this possibility, we use proteins from a different biologically validated non-pathogenic gram-positive bacterium to build the non-effector set. Considering the construction of training dataset from various species, it can be safe to say that the model developed will be a robust one and will not overfit. Thus the existence of some species-to-species variation in the features of the proteins can be taken care of.

A set of 2820 genes and their corresponding proteins of *Faecalibacterium prausnitzii* has been obtained from KEGG. Proteins annotated as “hypothetical”, and “putative” were removed from the set due to the unavailability of any functional information pertaining to them. Finally, a set of 1846 non-effector proteins of *Faecalibacterium prausnitzii* has been considered to form the non-effector set.

5.2.2 Feature extraction

In this section, we describe the features derived from the sequences of T7 effector and non-effector proteins. A schematic representation of the feature set is given in Figure 5.1.

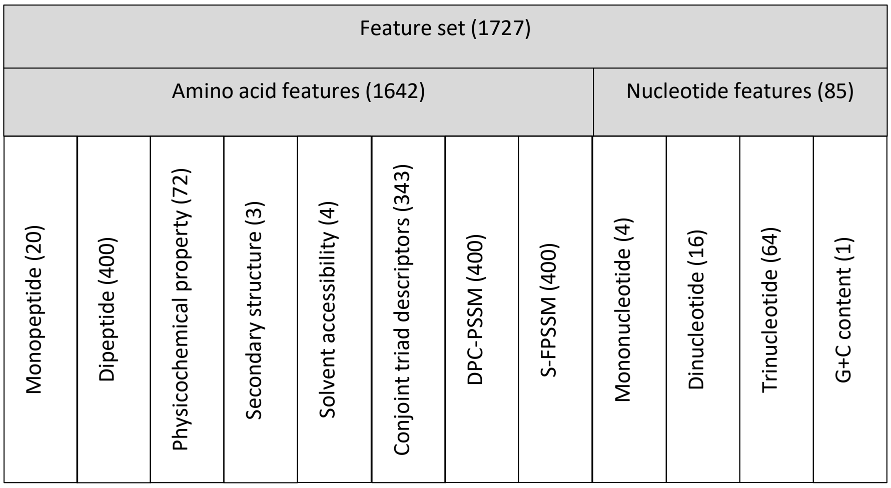


Figure 5.1: The diagram depicts the feature set for identification of T7 effector proteins. The number within "()" depicts the number of features generated from that category of feature. The feature set consists of 1727 features, comprising 1642 features based on amino acid sequences, and 85 features obtained from nucleotide sequences of the corresponding genes coding these effector proteins.

Features derived from amino acid sequences

Here, we have given a brief description of the features derived from the amino acid sequences. These features are related to amino acid frequency, physicochemical property, secondary structure, solvent accessibility, conjoint triad descriptors, and are those generated on evolutionary information.

Monopeptide and Dipeptide The percentage composition of 20 amino acids (A, G, V, E, I, L, P, F, Y, M, S, T, H, Q, N, W, K, R, C, and D) in a protein form the monopeptide sequence profile (MPSP). Thus we have 20 features corresponding to 20 amino acids. Similarly, the percentage composition of 400 di-peptides (AA, AG, AH, ..., and others) in the protein sequence form the dipeptide sequence profile (DPSP), which generates 400 features.

Physicochemical property We have considered 38 physicochemical properties from which 72 features have been extracted for the identification of T7 effector proteins. The various physicochemical properties and the spectrum of amino acids corresponding to them are tabulated in Table 5.1.

Let us consider Table 5.1. Features pertaining to the physicochemical properties 1 to 17, have been derived by calculating the percentage composition of the amino acids, given in (), known to possess that property. For example, consider the physicochemical property "aromatic". The feature value corresponding to the property "aromatic" is obtained by taking

Table 5.1: Summary of the features derived from physicochemical properties of proteins. Column "Properties" contains the name of the property, and column "Count" represents the number of features derived from the corresponding property. Total number of physicochemical features considered is 72.

No	Properties	Count	No	Properties	Count	No	Properties	Count
1	aromatic (F, H, W, Y) [413]	1	14	$1.0 <$ dipole < 2.0 (Y, M, T, S) [238, 277]	1	27	signal sequence helical potential and membrane-buried preference parameters [16]	2
2	charged (H, E, R, D, K) [413]	1	15	$2.0 <$ dipole < 3.0 (H, W, Q, N) [238, 277]	1	28	average flexibility index [39]	1
3	neutral (Y, G, S, P, H, T, A) [413]	1	16	dipole > 3.0 (R, K) [238, 277]	1	29	polarizability parameter and free energy of solution in water [74]	2
4	aliphatic (I, L, V) [413]	1	17	dipole > 3.0 with opposite orientation (D, E, C) [238, 277]	1	30	relative mutability [101]	1
5	polar (N, E, Q, K, R, D) [413]	1	18	hydrophobicity factor [156]	1	31	principal component I, II, III, IV [372]	4
6	hydrophobic (W, F, M, L, I, V, and C) [413]	1	19	residue volume [41]	1	32	normalized van der Waals volume and localized electrical effect [140]	2
7	transmembrane amino acid (A, L, V, I) [413]	1	20	transfer free energy to surface and apparent partial specific volume [55]	2	33	partition coefficient [152]	1
8	negatively charged (D and E) [413]	1	21	steric parameter [73]	1	34	hydration number [184]	1
9	positively charged (K, R, and H) [413]	1	22	average volume of buried residue [84]	1	35	entropy of formation, absolute entropy and heat capacity [196]	3
10	small (E, H, I, L, K, M, N, P, Q, and V) [413]	1	23	residue accessible surface area in tripeptide [85]	1	36	molecular descriptors [207]	22
11	tiny (A, T, D, S, G, and C) [413]	1	24	solvation free energy and atom-based hydrophobic moment [130]	2	37	refractivity [201]	1
12	large (F, W, R, and Y) [413]	1	25	molecular weight and melting point [139]	2	38	retention coefficients in HPLC [274]	2
13	dipole < 1.0 (A, G, V, I, L, F, P) [238, 277]	1	26	percentage of buried residues and percentage of exposed residues [200]	2			

sum of percentage composition of the individual amino acids F (phenylalanine), H (histidine), W (tryptophan) and Y (tyrosine) since these amino acids are aromatic. Feature values corresponding to the properties 2 to 17 are extracted in the same way.

Physicochemical properties 18 to 38 (Table 5.1) correspond to numerical values for 20 amino acids. For example, the property of "hydrophobicity factor" is represented by the values: A=0.75, L=2.4, R=0.75, K=1.5, N=0.69, M=1.3, D=0, C=1, F=2.65, P=2.6, Q=0.59, S=0, E=0, T=0.45, G=0, W=3, H=0, Y=2.85, I=2.95, and V=1.7. Value of the feature pertaining to the property of "hydrophobicity factor" is obtained by multiplying the amino acid values with the percentage composition of the corresponding amino acid in the protein sequences, followed by their sum. Likewise, the other feature values pertaining to the categories 18 to 38 are obtained as mentioned above. The values of amino acids with respect to each of the categories are furnished in Tables A.14 and A.15 of Appendix A. These 72 features form the physicochemical property profile (PPP).

Secondary structure Three properties based on the secondary structure of a protein have been taken into consideration. They are helix (H), coil (C), and sheets (E) to form the secondary structure sequence profile (SSSP) for a protein sequence. The amino acids L, A, E, M, K, Q, H, and R tend to form helix [307]. Likewise, the amino acids S, N, D, G, and P are known to form coil. Lastly, the amino acids Y, I, V, C, T, F, and W tend to collectively form sheet [298]. Higher the presence of amino acids pertaining to a particular secondary structure in a protein more is the chance of that protein to take the shape of that structure. The sum of the individual percentage compositions of amino acids known to form helix (H) represents the feature value for helix. In the same way, the sum of the individual percentage compositions of amino acids belonging to coil (C) and sheets (E) are considered to be the feature values representing coils and sheets respectively.

Solvent accessibility Solvent accessibility of an amino acid involves four properties, namely, very exposed (E), somewhat exposed (e), very buried (B), and somewhat buried (b). An amino acid is said to be very buried (B) when its accessibility is at most 4%, somewhat buried (b) when accessibility is between 4% and 25%, and somewhat exposed (e) when accessibility is between 25% and 50%. Likewise, an amino acid is called very exposed (E) when its accessibility is more than 50% [287, 334]. Amino acids that can be characterized as very buried are V, I, F, C, L, A; somewhat buried amino acids are Y, H, T, P, S, M, and W. Similarly, amino acids that are exposed are D, E, Q; and that of somewhat exposed are G, K, N, and R. The sum of the individual percentages of amino acids for a protein belonging to the property "very exposed (E)", is the feature value of the protein corresponding to that property. In the same way, the sum of the individual percentage composition of amino acids belonging to the properties somewhat exposed (e), very buried (B), and somewhat buried (b), are considered to be the feature values pertaining to these properties respectively. These four features have been considered to form solvent accessibility sequence profiles (SASP) of protein sequences.

Conjoint Triad Descriptors (CTD) The conjoint triad descriptors are extracted from the amino acid sequences. It is represented by a group of three consecutive amino acids (triads) in a protein sequence. The classification of these groups is based on the dipole scale of each amino acid and volumes of side chains [81]. The amino acid distribution pertaining to each of the groups has been given in Table 5.2. There are seven classes. Each amino acid falls into one of the seven types of classes. To find CTD, three consecutive amino acids (triplet) need to be considered. Considering a combination of three consecutive amino acids in a peptide sequence, each of the three amino acids will belong to one of the groups. The combination of the groups for three consecutive amino acids looks like [3, 1, 7], for example, if these three amino acids are in Groups 3, 1 and 7, respectively. As three consecutive amino acids have been taken into consideration and each amino acid can belong to a single group, there can be one of 343 $(= 7 \times 7 \times 7)$ possible groups for each triplet of amino acids. The frequency of triplets belonging to each of these 343 combinations of groups is considered to obtain 343 features (CTD). The frequency of each triad belonging to one of the combinations of groups forms the CTD. Let us consider an example of peptide sequence AMTSWP. The combinations of AMT, MTS, TSW and SWP are [1, 3, 3], [3, 3, 3], [3, 3, 4], and [3, 4, 2]. Hence, the frequencies of the combinations [1, 3, 3], [3, 3, 3], [3, 3, 4] and [3, 4, 2] are 1, 1, 1 and 1 respectively, while the rest of the combinations have frequencies of 0.

Table 5.2: Summary of the distribution of amino acids based on their dipole and volumes of the side chains

Group	Amino acids under each group
1	Alanine (A), Glycine (G), Valine (V)
2	Isoleucine (I), Leucine (L), Phenylalanine (F), Proline (P)
3	Tyrosine (Y), Methionine (M), Threonine (T), Threonine (S)
4	Histidine (H), Asparagine (N), Glutamine (Q), Tryptophan (W)
5	Arginine (R), Lysine (K)
6	Aspartic acid (D), Glutamic acid (E)
7	Cysteine (C)

Evolutionary information-based features A large number of studies have shown that evolutionary information is crucial in identification of effectors [409, 456]. Evolutionary information of a protein plays a crucial role in determining their functionality [64], and therefore, can serve as a basis for additional features to identify T7 effector proteins. The evolutionary information-based features considered are dipeptide composition-position specific scoring matrix (DPC-PSSM) and standardized filtered position-specific scoring matrix (S-FPSSM).

1. DPC-PSSM A position-specific scoring matrix (PSSM) of a protein is an l × 20 matrix, where l is the length of the amino acid sequence of a protein, and 20 represents the number of all the amino acids. A (k, j)th element of PSSM denotes the chance (represented as log likelihoods) of amino acid j to appear at the kth position of the protein sequence, such that $0 \leq k \leq l, 1 \leq j \leq 20$ [409]. The concept of dipeptide composition (DPC) position-specific scoring matrix encoding algorithm has been applied to generate PSSM, and to further generate DPC-PSSM from PSSM. An (i, j)th entry of DPC-PSSM can be calculated as:

$$Y = (y_{1,1}, \ldots, y_{1,20}, y_{2,1}, \ldots, y_{2,20}, \ldots, y_{20,1}, \ldots, y_{20,20})^T \tag{5.1}$$

such that

$$y_{i,j} = \frac{1}{l-1} \sum_{k=1}^{l-1} p'_{k,i} \times p'_{k+1,j}; 1 \leq i, j \leq 20 \tag{5.2}$$

where $p'_{k,i}$ and $p'_{k+1,j}$ denote the chances of amino acids i and j to appear respectively at the kth and $(k+1)$th positions (rows) of PSSM. DPC-PSSM is represented by the matrix Y of size 20 × 20, thus generating 400 features, which incorporate evolutionary information and reflect the sequence-order information [258]

2. S-FPSSM Standardized Filtered Position-Specific Scoring Matrix (S-FPSSM) is designed to extract evolutionary information-based on the matrix transformation of the original PSSM [444]. The filtered PSSM (fp) is generated from PSSM in a preprocessing step during which all negative elements (elements of PSSM being log likelihoods) of the PSSM are set to zero and all positive elements greater than an expected value δ' (with a default value of 7) are set to δ'. Consequently, all elements in FPSSM are in the interval of [0, δ']. Since the filtered matrix is obtained by filtering elements of PSSM, which is an l × 20 matrix, therefore filtered PSSM (fp) is also of size l × 20. Based on FPSSM, the resulting feature matrix $Y' = (y'_{1,1}, \ldots, y'_{1,20}, \ldots, y'_{20,1}, \ldots, y'_{20,20})$ can be defined as follows:

$$y'_{i,j} = \sum_{k=1}^{l} fp_{k,j} \times g; 1 \leq i, j \leq 20 \tag{5.3}$$

subject to the condition

$$
\begin{aligned}
g = 1 \text{ if } r'_k = r''_i; i, j = 1, \ldots, 20 \\
g = 0 \text{ if } r'_k \neq r''_i; i, j = 1, \ldots, 20
\end{aligned}
\tag{5.4}
$$

where l denotes the length of the protein sequence; $fp_{k,i}$ denotes the element in the kth row and ith column of FPSSM; r'_k stands for the kth residue in the sequence, and r''_i represents the ith amino acid of 20 primary amino acids. S-FPSSM is represented by the matrix Y of size 20 × 20, thus generating 400 features.

Features derived from nucleotide sequences

These features have been derived from the nucleotide sequences of the genes. The percentage composition of four mononucleotides (A, T, G, C) in a gene, i.e., the percentage of each of A, T, G and C with respect to the total number of nucleotides in the sequence of the gene form mononucleotide sequence profile (MNSP). Likewise, the percentage composition of 16 dinucleotides (AA, AT, AG, ..., and others) with respect to the total number of dinucleotides in the gene sequence form the dinucleotide sequence profile (DNSP). The percentage composition of 64 tri-nucleotides (AAA, AAT, AAG, ..., and others) with respect to the total number of triplets form trinucleotide sequence profiles (TNSP). Thus nucleotide sequence profile (NSP) of a gene comprises MNSP (4 features), DNSP (16 features), TNSP (64 features), and GC content (1 feature). In this way, we have got 85 features for a gene corresponding to a protein.

5.2.3 Preprocessing of feature set

In order to train a classifier model, the original dataset is divided into training and test sets, in the ratio of 7:3. As observed, the T7 training and test datasets are unbalanced, *i.e.*, the number of samples in the effector class is considerably less than that in the non-effector class. Equal sized sets of effector and non-effector proteins need to be considered for training purposes, in order to avoid unequal class distribution which eventually results in a biased classifier [150]. In order to do so, we have oversampled the training dataset using Cluster Quality-based Non-Reductional (CQNR) oversampling algorithm [355], recently developed by the authors, so that cardinality of the minority class (T7 effector proteins) becomes approximately equal to that of the majority class (non-effector proteins).

CQNR is chosen for oversampling due to its capability to handle high dimensional data, generation of synthetic samples by maintaining class distribution, and non-removal of samples as noise. CQNR retains the properties of the original dataset and generates the minimum number of synthetic samples required for balancing majority and minority classes; consequently maintaining the distribution of the original dataset. It can handle oversampling of disjoint clusters of data points of the minority class. It does not eliminate any sample as noise. To ensure maximum resemblance of the generated synthetic samples to the experimentally validated data, CQNR generates the samples in the vicinity of the experimentally verified samples without replicating the original samples themselves. As a result, CQNR generates synthetic samples by taking into account the data distribution in the minority classes, thereby effectively reducing the bias introduced by class imbalance. For the above-mentioned characteristics, CQNR has been chosen over other oversampling algorithms. Oversampling of the unbalanced dataset is followed by standardization of the features by subtracting mean values from them followed by scaling them to unit variance.

When a dataset is divided into training and test sets, the distribution of the samples in the test set should be similar to that of the training set in order to develop a robust model for classification/prediction. On the other hand, if the distributions of training and test samples are different, the performance of this model on the test set is misleading. This phenomenon is referred to as covariance shift [383].

In order to identify the existence of covariate shift, we treat the samples of training set belonging to one class (say class 0) while the samples of test set belonging to the other (say class 1). We form a new dataset with samples from the training and test set of effector protein dataset. Let us assume that the distributions of training and test sets are different, i.e., covariate shift exists. Therefore, samples with label 0 are easily differentiable from the samples labeled 1. Since classes 0 and 1 are differentiable, classifiers would give a high performance for the new dataset. However, if covariance shift does not exist, it would mean that the distributions of the training and test sets are nearly the same. Consequently, samples with label 0 are not easily differentiable from the samples labeled 1. Hence, classifiers trained on the new dataset would result in low values of performance measures. Thus, to detect whether there is covariant shift between training and test sets, a classifier trained on samples belonging to the classes of training and test sets must result in low value of performance parameter, say MCC. If MCC value is greater than 0.2, it can be said that there is a covariate shift and the division of training and test datasets of DeepT7 needs to be redone [383]. However, if MCC value is less than 0.2, it is safe to conclude that there is no covariate shift. In the present study, we have performed the procedure mentioned above to detect covariance shift. The performance of a classifier (MCC=0.0243) on the dataset considered by DeepT7 has ruled out the possibility of the existence of covariate shift in the dataset.

The feature set has been further subjected to feature selection to avoid overfitting of the classifiers. We have used univariate feature ranking method to rank the features according to their importance [202]. Thus, we have got subsets consisting of the top 5, 10, 15, 20, 25, . . .

100 percentage of features for classification. In order to avoid overfitting, only the top 5% highly important features have been used for further classification and prediction.

5.2.4 Architecture of DeepT7

The performance of multiple learning models has been analyzed to derive a suitable model for the classification of T7 effectors. The performance of DeepT7 has been compared against various models considered, viz., support vector machine (SVM), k-nearest neighbor (k-NN), decision tree (DT), naive-Bayes (NB) and random forest (RF). The hyperparameter values of the classifiers are given in Section A.3.2 of Appendix A. DeepT7 has reported better performance with respect to these models, as depicted in Table 5.3. Additionally, an ensemble model involving these six models has been considered. This ensemble model generated a better performance compared to the individual models, except DeepT7. DeepT7 has reported a better classification performance compared to the ensemble model considered.

There is a possibility that an individual model performs better than an ensemble model [178,443]. For example, consider five classifiers forming an ensemble model where the final class is decided by majority voting. Suppose for a certain sample, the actual class label is 0. However, two of the classifiers predict the sample to belong to class 0 while the other three classifiers predict it to belong to class 1. In such a case, the ensemble model would predict the label of the unknown class to be 1. Therefore, the ensemble model is said to have misclassified the sample. This has been the case for the ensemble model and DeepT7. It has been noticed that DeepT7 has predicted the correct class labels of the samples of the test set which the ensemble model has not. This indicates clear supremacy of DeepT7 over the other classifiers in terms of performance, including the ensemble model. Thus we have considered DeepT7 here for classification of T7 effector proteins. A detailed comparison of DeepT7 with the individual models has been furnished in Section 3.

The deep neural network framework with dropout regularization [178] utilized in DeepT7 consists of two hidden layers with $ReLU$ as the activation function, along with the input and output layers. The output layer has a sigmoid activation function. It has been assumed that class label 0 is used to determine an effector and class label 1 has been assigned to non-effector. Therefore, if the predicted value for an unknown protein is within the interval [0.5,1], it is identified as a non-effector. Otherwise, the protein is predicted as an effector.

In dropout regularization, some of the input and hidden units are randomly neglected during training to prevent their co-adaptation [29]. Dropout regularization has been applied to DeepT7 since it reduces overfitting and improves the generalization capability of DeepT7 [377]. Since effector identification is a binary classification problem, cross-entropy has been selected as the loss function. We have chosen Adam optimizer for our model. Adam is an optimization algorithm for training deep learning models. Adam combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems [217]. Parameter tuning has been used to optimize a set of important hyper-parameters, *viz.*, learning rate, batch size, maximum epoch and early stopping patience.

DeepT7 contains two hidden layers with the number of nodes being 50 and 25 respectively. We have set the learning rate at 0.005, momentum factor at 0.9, the maximum number of training epochs at 100, and early stopping patience at 10. The optimal value of the batch size tuned on the cross-validation set is 30. The flow of DeepT7 has been depicted in Figure 5.2. In order to prevent overfitting and for parameter tuning, 10-fold cross-validation has been implemented. The labeled training data have been partitioned into 10 non-overlapping equal-sized sets, and the model has been trained on the union of nine of these sets before

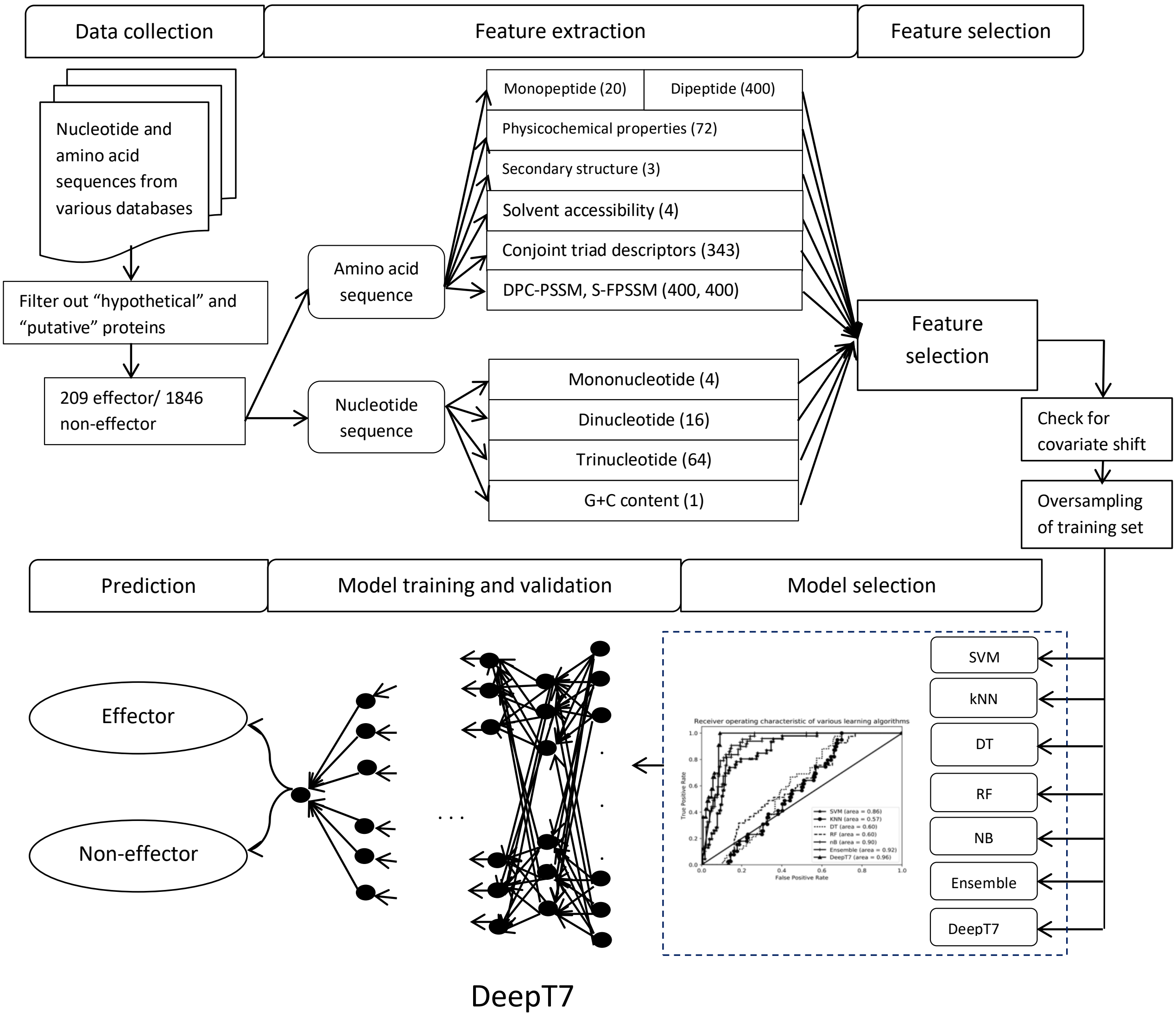


Figure 5.2: The diagram depicting the flow of DeepT7.

being tested on the remaining one. This has been repeated 10 times, such that each of the 10 sets is used as the test set exactly once, and the average performance parameters have been recorded. DeepT7 has further been subjected to holdout testing. This method of testing a model, also known as independent testing, has no common samples between the training and test sets [223]. DeepT7, after being validated by 10-fold cross-validation, has been subjected to holdout testing on the test set, and the values of performance measures have been recorded.

5.3 Results

DeepT7 uses a deep neural network framework to decide whether an unknown protein is a T7 effector or not. We have compared the performance of deep neural network for classification of T7 effectors against the performance of support vector machine, k-nearest neighbor, decision tree, random forest, naive Bayes and an ensemble model. It has been noticed that the deep neural network model has performed better compared to the other classifiers. Since no investigations exist in identification of T7 effectors, a quantitative comparative analysis of our system could not be provided.

5.3.1 Performance evaluation

Multiple learning models have been tested for classification of T7 effectors. We have compared the performance of support vector machine (SVM), k-nearest neighbor (k-NN), decision tree (DT), naive-Bayes (NB), random forest (RF) and ensemble model with DeepT7. DeepT7 has reported the best performance compared to the other models. A summary of the performance of various classifiers has been furnished in Table 5.3. For two-class classification problems, seven performance measures, namely, *Accuracy (ACC), Sensitivity (SN), Specificity (SP),* F-score, G-mean, Cohen's κ score and *MCC*, have been used to evaluate the overall predictive performance of classification models. The receiver operating characteristic (ROC) curve has been plotted to visually measure the performance of different methods. The ROC curve of DeepT7 compared to the other models has been provided in Figure 5.3. The area under the curve (AUC) is also provided in each of the ROC plots. As observed from Table 5.3, the performance of DeepT7 supersedes the performance of other models for most of the performance measures. DeepT7 has reported an *accuracy* of 91.50%, *sensitivity* of 91.12%, *specificity* of 99.14%, *F-score* of 0.6721, *G-mean* of 0.9504, κ-score of 0.6467 and an *MCC* of 0.7480. The next best performance is by the ensemble model with an *accuracy* of 89.53%, *sensitivity* of 86.92%, *specificity* of 81.24%, *F-score* of 0.6624, *G-mean* of 0.8254, κ-score of 0.6783 and with an *MCC* of 0.7034. However, the random forest classifier has

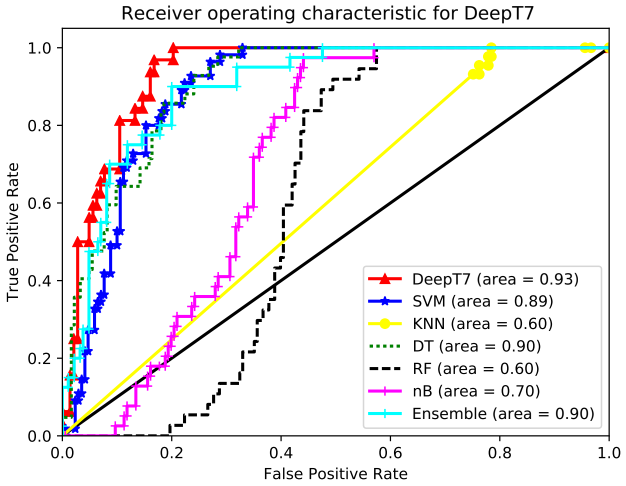
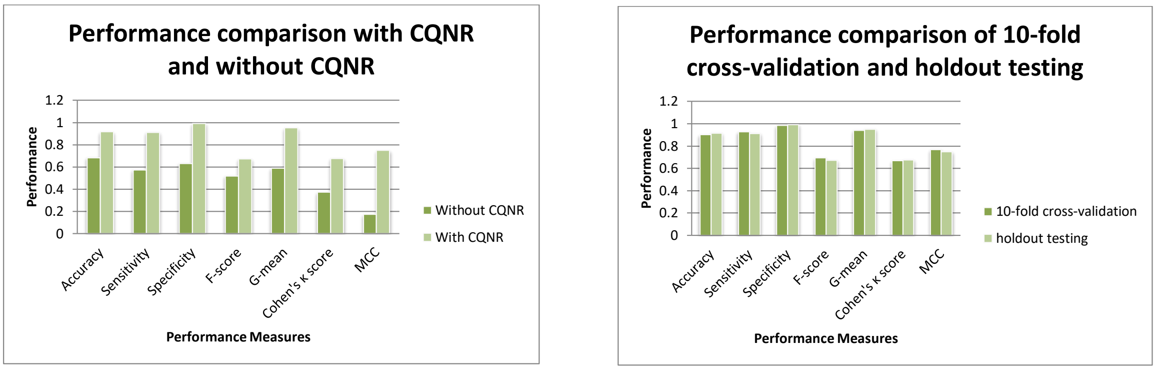


Figure 5.3: Comparison of Receiver Operating Characteristic (ROC) of six other classifiers with DeepT7

reported a maximum Cohen's κ score of 0.6836 among the other classifiers.

A comparison of the performance of DeepT7 with and without the oversampling algorithm CQNR has been depicted in Figure 5.4 (a). As observed, application of CQNR has significantly improved the performance of DeepT7. Comparison of the performance of DeepT7 with respect to 10-fold cross-validation and holdout testing has been depicted in Figure 5.4 (b). It has been noticed that the difference in performance of both the methods is negligible, indicating DeepT7 to be a robust and effective T7 effector protein identifier.



(a) Performance comparison of DeepT7 with and without CQNR

(b) Performance comparison of DeepT7 for 10-fold cross-validation and holdout testing

Figure 5.4: Comparison of the performance of DeepT7 with respect to oversampling and testing.

Table 5.3: Summary of performance of the five classifiers with 10-fold cross-validation. The tabulated values are the 50-fold average for each of the classifiers. The maximum value for every performance measure has been highlighted.

Classifier	*Accuracy* (%)	*Sensitivity* (%)	*Specificity* (%)	F-score	G-mean	Cohen's κ score	*MCC*
DeepT7	**91.50**	**91.12**	**99.14**	**0.6721**	**0.9504**	0.6467	**0.7480**
Support vector machine	84.24	80.27	81.35	0.5586	0.8080	0.6192	0.6023
k-Nearest Neighbors	86.35	83.76	79.53	0.6276	0.8161	0.6534	0.6621
Decision tree	82.38	81.34	83.14	0.5984	0.8223	0.6693	0.6528
Random Forest	85.35	81.45	78.45	0.5269	0.7993	**0.6836**	0.6843
Naive Bayes	88.23	84.12	79.23	0.6129	0.8183	0.6623	0.6918
Ensemble model	89.53	86.92	81.24	0.6624	0.8254	0.6783	0.7034

5.3.2 Application of DeepT7 on proteins of *Mycobacterium bovis* and *Streptococcus pneumoniae*

We have applied DeepT7 on the entire genome of two gram-positive, pathogenic organisms, namely, *Mycobacterium bovis* and *Streptococcus pneumoniae*. The genomes were downloaded from KEGG and contained both amino acid and their corresponding nucleotide sequences. As in July 2020, *M. bovis* had 1966 amino acid and their corresponding nucleotide sequences of which 45 were predicted to be T7 effector proteins by DeepT7, while in *S. pneumoniae* the algorithm found 39 out of 2125 sequences to be T7 effectors. Analysis of each predicted protein, based on the biological process they are involved in, the cellular component in which they reside and molecular function evidence along with their prediction probabilities, have been summarized in the file 'biological_validation.xls' under the 'Biological validation' section of our website.

Predicted probable effector proteins in *Mycobacterium bovis:* Among the 45 proteins predicted to be T7 effectors, eight are ESX proteins and five are hypothetical. All ESX proteins have biological functionality of pathogenesis. Thus, we can consider them to be effectors [364]. The functionality of the five hypothetical proteins has not yet been explored and documented in literature, and therefore leaves these proteins a chance to be effectors. Protein diacylglycerol O-acyltransferase has not been reported to cause pathogenesis. However, it is secreted and resides in the extracellular location, and thereby indicating a chance of the protein to be a T7 effector [280]. Proteins PE18, PE19 and PE35 from PE family-related proteins are secretion peptides, and have shown a 100% sequence similarity with the predicted eight ESX proteins, indicating that these can be T7 effectors [354]. However, their functionality has not yet been determined by experimental results. Twenty two proteins belonging to the PE-PGRS protein family have found their place in the predicted T7 effector list. Interestingly, it has been experimentally validated that these proteins are indeed secreted by the pathogen *Mycobacterium tuberculosis* into the host, indicating its contribution to pathogenesis [280]. The antitoxin protein from gene MAZE7 has been predicted to be a T7 effector. Antitoxin in pathogenic bacteria is known to assist in bacterial growth, and therefore indirectly working in support of pathogenicity [243].

Predicted probable effector proteins in *Streptococcus pneumoniae:* Among the 39 proteins predicted as effectors by DeepT7, six either confirmed to be T7 effectors or have some chance of being T7 effectors since they are annotated to be pathogenic. These six proteins are known to perform other functions, like participation in glycolytic process, cell adhesion, biofilm formation and participation in its metabolic processes apart from pathogenesis. Three of these predicted proteins are bacteriocins, which are toxins produced by bacteria to inhibit the growth of similar or closely related bacterial strains [25]. Nine transport system proteins have been predicted to be T7 effectors. Biologically, these proteins are known to promote pathogenesis in various organisms [387]. These nine transport system proteins are known to perform various functions, like ATP binding, transmembrane transporter activity and DNA binding. Among the transport proteins, two of them have a defined biological function of pathogenesis, apart from the other functions mentioned above. Seven cell wall proteins, also known to assist the pathogenic nature of the organism [107], have been predicted to be T7 effectors. Two hypothetical proteins have also been predicted to be T7 effectors.

5.3.3 Analysis of DeepT7 with respect to other effector protein predictors

Due to the absence of T7 effector protein identification models in literature, a qualitative comparison of DeepT7 is provided against predictors of other types of effector proteins. There exist algorithms for identification of T3 effector proteins [17, 261, 338, 433, 440], T4 effector proteins [59, 413, 432, 456], and a limited number of algorithms for identification of T6 effector proteins [353, 409]. The comparison of the models is given below.

Comparison with T3 effector protein predictors: Identification of secretion signals in T3 effector proteins of gram-negative bacteria using an ANN (Artificial Neural Network) with gradient descent back-propagation learning with momentum and SVM (Support Vector Machine), has been done by Löwer *et al.* [261]. While creating the effector/non-effector dataset, sequences with fewer than 100 amino acids were removed. This may have led to a loss of information since small protein sequences can be pathogenic [264]. However, DeepT7 has considered all available proteins for generation of the feature set. Only 575 features have been taken into consideration by Löwer *et al.* for effector identification. However, DeepT7 has considered 1727 features, thus leading to the development of a more robust and accurate model. The training performance of the classifiers has been measured by *MCC*, while DeepT7 has used seven performance measures (*accuracy, sensitivity, specificity, F*-score, *G*-mean, κ*-score* and *MCC*).

Samudrala *et al.* [338] have used SVM to design a tool called SIEVE, to predict the T3 effectors. This investigation has used five groups of features without clearly stating the content of each group. DeepT7 has generated 12 groups of features resulting in a total of 1727 features (Figure 5.1). Unlike DeepT7, *sensitivity* and *specificity* are the two performance measures used to measure the efficiency of SIEVE. Yang *et al.* have used SVM on features extracted from N-terminal of T3 effector proteins for their identification [440]. DeepT7, on the other hand, has considered all amino acids from the effectors. The training dataset has been constructed from the whole genome of *Pseudomonas syringae*, and therefore, the predictor is not generalized. DeepT7, on the other hand, has a much more generalized training dataset, consisting of T7 effector proteins from various organisms. The investigation has considered 160 features, while DeepT7 has considered a more diverse feature set.

Identification of T3 secreted proteins has been made based on the amino acid sequences by Arnold *et al.* [17]. Unlike considering the whole amino acid sequence of effector proteins for feature extraction as in the case of DeepT7, this investigation has considered the N-terminal amino acids of effector proteins, which has led to 70 features; thus losing a considerable amount of information that could have been obtained from the whole protein. The investigation has used *sensitivity, specificity* and *AUC* to measure its performance. DeepT3, developed by Xue *et al.* [433], has used the first 100 residues of T3 effector proteins, in a deep learning framework to predict T3 effector proteins. This investigation too, unlike DeepT7, has extracted features from the N-terminal of proteins, which is limited to 100 amino acids.

Comparison with T4 effector protein predictors: Zou *et al.* [456] have reported an SVM classifier to identify T4 effector proteins based on amino acid composition. This investigation has considered a total of 440 features, considerably fewer than the feature set considered by DeepT7. Even though the dataset was heavily unbalanced, no oversampling or under-sampling technique has been used by this investigation. For measuring the performance of their classifier, the measurements *accuracy, sensitivity, specificity* and *MCC* have been used. It has been noticed that specificity of the method ranges from 90% to 98%, while sensitivity ranges from 53% to 77%. As observed, the method has a substantially high variance among the measurements, indicating quite an unstable performance. DeepT7 has reported a stable performance; the variance among the measurements is negligible. It has been noticed that the difference in performance of both methods is negligible.

Burstein *et al.* [59] have also made an attempt to predict T4 effectors. In *Legionella pneumophila*, the ORFs of proteins, instead of the whole protein, has been used by Burstein *et al.* to identify T4 effector proteins by using a machine learning approach [59]. Burstein *et al.* have used an ensemble learning of four classifiers to predict the class of an unknown protein by majority voting. However, they have not mentioned how a tie would be resolved, such that two classifiers predicted one class while the other two predicted the other class. This method has used *accuracy* and *AUC* to measure the performance of the classifiers.

Xiong *et al.* [432] and Wang *et al.* [413] have predicted T4 effectors using ensemble classifiers exclusively based on C-terminal features. The investigation by Xiong *et al.* [432] has considered only PSSM profile (400 features) as their feature set using which they have made further predictions, leaving out numerous crucial features considered by other predictors. The investigation did not report any precautionary measures taken to tackle data imbalance, even though their training data were imbalanced. This investigation has used ensemble learning with eight classifiers, yet has not reported how the final class label for a protein is assigned when there is a tie among these eight classifiers. Wang *et al.*'s investigation has led to the development of a T4 predictor called Bastion4. However, it has built a system that can predict effectors and non-effectors only in *Helicobacter pylori*. The issue of data imbalance has been ignored in this investigation.

Comparison with T6 effector protein predictors: Bastion6, an SVM-based T6 effector protein predictor, and PyPredT6, an ensemble learning-based T6 effector protein predictor, are currently the two available tools for the identification of T6 effector proteins [353,409]. Bastion6 has a considerably high variance among the measurements, indicating quite an unstable performance. This might have stemmed from the fact that Bastion6 has not applied any feature selection technique over its 1096 features. Bastion6 cannot be executed on protein sequences of length beyond the range 50-5000. The question of an unbalanced training dataset was also not addressed in that investigation, and therefore indicating a biased classifier. Both PyPredT6 [353] and Bastion6 have a smaller feature set of 873 features and 1096 features respectively, compared to DeepT7.

5.4 Conclusions

Identification of effector proteins from bacterial proteome is an important task for the analysis of the role of secretion systems in pathogenesis. It is the first step towards developing a cure for pathogenic diseases. Here we have developed a deep neural network-based system, called DeepT7, for the identification of probable T7 effector proteins. DeepT7 extracts a feature set containing 1727 features from nucleotide and amino acid sequences of experimentally verified T7 effector proteins, and based on these features predicts whether an unknown protein is a T7 effector or not. DeepT7 has predicted 45 out of 1966 proteins from *Mycobacterium bovis* and 39 out of 2126 proteins from *Streptococcus pneumoniae* to be T7 effectors. We have analyzed these predicted proteins with respect to their biological function and cellular location.

However, the investigation can be improved in the future. Firstly, the identification of effectors can be improved by incorporating 3D structural features of T7 effector proteins. Due to their unavailability in the current scenario, this investigation could not be carried out in this chapter. With time, when the 3D structural information for a substantial number of T7 effectors gets discovered, a more potent prediction system for T7 effector identification can be developed. Secondly, a more detailed biological validation for each putative candidate protein is essential, which forms a scope for further study. Finally, the methodology can be extended to other pathogens whose genomes and proteomes are either partially or entirely mapped.

So far we have worked towards identifying toxins liberated from pathogens, with the help of feature extraction and classification techniques. However, identification of toxins remains incomplete without studying their effect on host pathways. Thus, in Chapters 6 and 7, we delve into analyzing the effect of toxins, liberated by pathogens, on metabolic and signaling pathways of the hosts, relying on the concept of pathway prediction.

Chapter 6

ASAPP: Architectural Similarity-based Automated Pathway Prediction System and Its Application in Host-Pathogen Interactions [356]

In the last three chapters, we have dealt with identification of toxins, popularly known as effector proteins, secreted by gram-negative and gram-positive bacteria. However, the information on the mechanism by which toxins disrupt the host pathways, is crucial in designing possible therapies for a disease. Toxins disrupt both metabolic as well as signaling pathways. Release of toxins into the surrounding environment, regardless of when it was released, results in the disruption of metabolic pathways in the host eukaryote. Disruptions of these metabolic pathways include damaging cell membranes, disrupting protein synthesis, or inhibiting neurotransmitter release. For example, in Tuberculosis, the disease caused by the pathogen *Mycobacterium tuberculosis*, effector proteins enter into host macrophages and disrupt the metabolic pathways[1] [46]. In particular, the pathway forming ATP in the host is perturbed [360]. ATP generation is an extremely important function for survival of cells. Thus, it is of utmost importance that we investigate the effect of such toxins on various metabolic and signaling pathways in hosts. In this chapter, we attempt to study the effect of toxin on metabolic pathways.

Pathogens are infectious agents that disrupt the proper functioning of the host and cause diseases. One of the modus operandi by which pathogens ambush the host is via protein secretion, using the mechanism of secretion systems [354]. These secretion systems discharge effector proteins into the body of the host which have the capability to distort the usual metabolic pathways leading to the occurrence of unfamiliar transformations. Among

[1] https://www.genome.jp/kegg-bin/show_pathway?map05152+C01673

various toxins, other than effectors, small molecules are also secreted by pathogens into the host. These molecules can cause disease on contact with or absorption by body tissues interacting with biological macromolecules. This results in perturbation of the host system [305]. The significance of pathway prediction is to comprehend the possible undisclosed transformation(s) (reaction(s)) that can materialize provided the appropriate enzymes are available. Our algorithm is an attempt towards achieving this goal.

Multiple attempts have already been made for pathway prediction. *In silico* prediction of pathway came into existence when Karp *et al.* developed the PathoLogic tool [212], followed by the PathMiner [272], Pathway-Hunter [324], algorithm developed by Oh *et al.* [299], PathPred [285], and UM-PPS [132] predicting xenobiotic biodegradation pathways, and Rahnuma [282]. The mechanism behind the PathoLogic algorithm was hard-coded, with complicated interactions among various rules, making the algorithm difficult to maintain and extend. Following PathoLogic Tool, McShan *et al.* developed PathMiner [272], a heuristic-based path inferring algorithm. SMILES representation of chemical compound was used to represent metabolites in PathMiner [272] and Pathway-Hunter [324]. However, SMILES representation lacks a standard methodology to generate the representation. Canonical SMILES attempted to alleviate this issue, but there could be some variance in canonical SMILES depending on what tool was used to create them. For each canonical SMILES string of length n, there are $(n \times (n+1))/2$ different sequence of atoms [384]. Different representation of SMILES of the same metabolite leads to different similarity scores between two metabolites.

Similarly, PathMiner [272] uses Manhattan distance between the SMILES sequences of all the metabolite pairs to determine the similarity between them, thus predicting transformations among the metabolites. However, this method predicts a linear pathway without considering the possibility that branching in the pathway may exist. Likewise, InChI format-based software may generate different InChI strings for the same molecule, depending on the choice of a multitude of options [186]. It also lacks the ability to represent polymers. Pathway Hunter tool aims to find the minimum pathway between two metabolites. Soon after PathMiner, specialized tools like PathPred [285] and UM-PPS [132] attempted to predict only the xenobiotic biodegradation pathway. In reality, the metabolic pathways are not restricted to xenobiotic pathways. In fact, xenobiotic pathways make up for only 12% of the metabolic pathways (there are 181 pathways listed in KEGG, among which 21 are xenobiotic). Oh *et al.* and PathPred [285] used RDM (R: Reaction center; D: Difference atom; M: Matched atom) patterns for pathway prediction. In xenobiotic pathways, 80% of the RDM patterns corresponding to each of the transformations in a pathway is unique [299], and could be used to uniquely identify a transformation pair. Thus, the rule for transformation of one metabolite to another is more certain for the xenobiotic metabolite, provided the

RDM patterns were taken into consideration. Similarly, UM-PPS [132] [151] has been solely applicable for the prediction of bio-degradation pathways. It has a predefined set of transformation rules which needs to be manually updated in order to upgrade the algorithm. Another pathway prediction system, known as Rahnuma [282], used the existing experimentally verified reactions to create a pathway. It has consciously overlooked a set of metabolites and assumed an upper threshold value for the length of the pathways, above which the pathways were not taken into account.

In this chapter, we have designed a novel generalized algorithm, called Architectural Similarity-based Automated Pathway Prediction (ASAPP) which is used to predict pathways based on the structural resemblance of the metabolites. It has been seen that in a considerable number of pathways, there is structural similarity among the primary metabolites. ASAPP is a versatile algorithm which considers two-dimensional structure (atoms and bonds as well as molecular weight) of the metabolites, as inputs to build an array of probable transformations independently. It does not depend on any externally established reactions. Moreover, ASAPP has an accuracy of 85.09% when tested on 41 predefined pathways. We have applied the algorithm in the domain of host-pathogen interactions to analyze the effect of toxins on the metabolic pathways of the host. The implementation of the algorithm ASAPP has been made available at `http://asapp.droppages.com/`.

6.1 Method

In this section, we describe the developed methodology for automated pathway reconstruction. A pool of metabolites has been considered as input in the form of atoms and bonds as well as molecular weight. The output is a list of probable transformations in the form of compound pairs, indicating that the transformation between these two compounds are highly probable. We have extracted structural information of the metabolites from the KEGG database [209]. KEGG has been considered as the primary database due its versatility, routine updation and robustness (Section A.4.1 of Appendix A). Consider for example, a pathway given in Figure 6.1.

It is the oxidative phase of the pentose phosphate pathway, where Glucose 6P (C01172[2]) is the initial metabolite and Ribulose 5P (C00199) is the final metabolite. The arrows indicate the transformation of metabolites via the reactions[3]. For example, the metabolites D-Glucono-1,5-lactone 6-phosphate (C01236) and 6-Phospho-D-gluconate (C00345) are transformable via the reaction R02035. Using the present Architectural Similarity-based Automated Pathway Prediction (ASAPP) algorithm, we have computed the chance of occurrence

[2] Each metabolite in KEGG is identified by its unique ID of the format C*****.

[3] Each reaction in KEGG is identified by its unique ID of the format R*****.

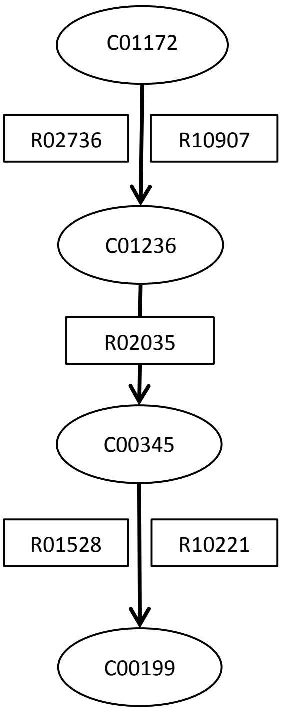


Figure 6.1: The oxidative phase of the Pentose Phosphate Pathway. The ovals contain the metabolite IDs and the rectangles stand for reactions. For example, metabolite beta-D-Glucose 6-phosphate (C01172) gets transformed into D-Glucono-1,5-lactone 6-phosphate (C01236) via the reactions R02736 and R10907 (as given in KEGG).

of these transformations of one metabolite to another, depending on the two-dimensional structural similarity between the metabolites.

6.2 Algorithm

The algorithm ASAPP has been designed to predict a pathway involving possible reactions among metabolites, based on two-dimensional structural similarities between a pair of metabolites. Each metabolite has been perceived as a undirected graph containing bonds and atoms as shown in Figure 6.2. The symbols used in the algorithm and their meaning have been summarized in Table 6.1. The modulated flow of ASAPP has been depicted in Figure 6.3.

6.2.1 Reading metabolite information from KEGG

Atoms, bonds among the atoms and molecular weights of the metabolites has been automatically extracted from KEGG by the algorithm. The algorithm reads the metabolite names as input and maps a name to a KEGG ID. Every metabolite in KEGG is associated with a

Table 6.1: Description of symbols used in ASAPP

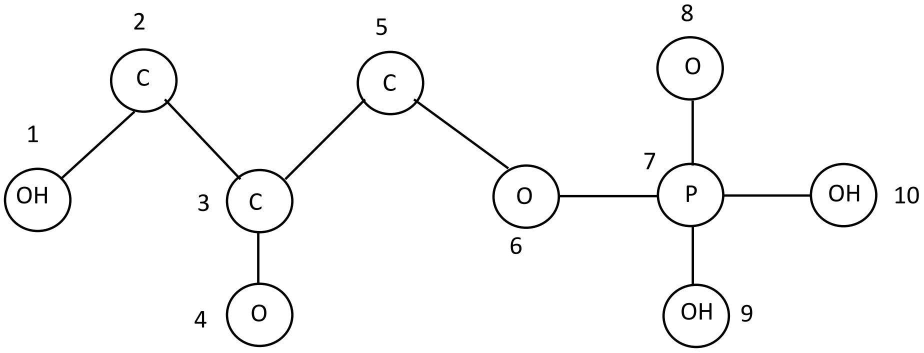
Symbol	Description
a $(\in \mathbb{N})$	Number of metabolites
a'_i $(\in \mathbb{N})$	Number of atoms in i^{th} metabolite
Δ_i $(\in \mathbb{N})$	Number of bonds in i^{th} metabolite
E_{ik}	k^{th} atom of i^{th} metabolite
M_n	Set of n input metabolite names obtained from KEGG
T	Set of metabolite pairs
$X_p^{(3)}$, $X_q^{(5)}$, $X_r^{(7)}$	Sets of atoms involved in p^{th}, q^{th} and r^{th} segments of length three, five and seven, i.e., each segment consisting of three, five and seven atoms respectively
$X_p'^{(3)}$, $X_q'^{(5)}$, $X_r'^{(7)}$	Sequence of atoms in the p^{th}, q^{th} and r^{th} segments of length three, five and seven, i.e., each segment consisting of three, five and seven atoms respectively
$X_i''^{(3)}$, $X_i''^{(5)}$, $X_i''^{(7)}$	Sets of segments of length three, five and seven, generated from ith metabolite
$\epsilon_{ij}^{(3)}$, $\epsilon_{ij}^{(5)}$, $\epsilon_{ij}^{(7)}$ $(\in \mathbb{N})$	The number of common three, five and seven-atom segments, respectively, between i^{th} and j^{th} metabolites
$\epsilon_{ij}'^{(3)}$, $\epsilon_{ij}'^{(5)}$, $\epsilon_{ij}'^{(7)}$ $(\in \mathbb{R})$	Standardized score of the number of common three, five and seven-atom segments, respectively, between i^{th} and j^{th} metabolites
$\omega''_{i,j}(\in \mathbb{R})$	Summation of $\epsilon_{ij}'^{(3)}$, $\epsilon_{ij}'^{(5)}$, $\epsilon_{ij}'^{(7)}$
w_i $(\in \mathbb{R})$	Molecular weight of i^{th} metabolite
ω'_{ij} $(\in \mathbb{R})$	Standardized difference in molecular weight between i^{th} and j^{th} metabolites
ω_{ij} $(\in \mathbb{R})$	Final score depicting the similarity between i^{th} and j^{th} metabolites
$\mathscr{C}_i^{(1)}$, $\mathscr{C}_i^{(2)}$, $\mathscr{C}_i^{(3)}$	Sets of metabolites/compounds having highest, second highest and third highest similarity score values, respectively, with i^{th} metabolite

unique KEGG ID. A list of KEGG IDs and the corresponding names of a metabolite[4], has been formed. The algorithm uses this list to map a metabolite name to its respective KEGG ID. For each metabolite, the corresponding two-dimensional structure, in the form of atoms and bonds, has been obtained on-line from the KEGG KCF (Appendix B.3) files, along with its molecular weight. Using this information, the process of segmentation of metabolite has been carried out.

6.2.2 Segmentation of the metabolites

After accumulation of information, the next stage is segmentation. In a reaction, product metabolite have been formed by integrating multiple segments of two or more reactants. Segments are continuous linear sequence of connected atoms, such that an a'-atom sequence has $a'-1$ bonds. Three, five or seven-atom segments have been considered for representing a

[4] Citric acid is identified by KEGG ID C00158. It is also referred to as Citrate, 2-Hydroxy-1,2,3-propanetricarboxylic acid and 2-Hydroxytricarballylic acid.



Edges	3 sized segment	Atom format	5 sized segment	Atom format	7 sized segment	Atom format
1-2	1-2-3	OCC	1-2-3-5-6	OCCCO	1-2-3-5-6-7-8	OCCCOPO
2-3	2-3-4	CCO	2-3-5-6-7	CCCOP	1-2-3-5-6-7-10	OCCCOPO
3-4	2-3-5	CCC	3-5-6-7-8	CCOPO	1-2-3-5-6-7-9	OCCCOPO
3-5	3-5-6	CCO	3-5-6-7-10	CCOPO		
5-6	5-6-7	COP	3-5-6-7-9	CCOPO		
6-7	6-7-8	OPO				
7-8	6-7-9	OPO				
7-9	6-7-10	OPO				
7-10	8-7-9	OPO				
	8-7-10	OPO				
	9-7-10	OPO				

Figure 6.2: Two-dimensional structure of the metabolite Glycerone phosphate (C00111) has been laid out as given in KEGG KCF (XML format) files, where each atom has been numbered. Segments of length three, five and seven have been constructed, and their constituent atoms have been shown. The edges represent the bonds between the atoms.

metabolite. Some metabolites are so small that a five or seven-atom segment cannot be used represent the metabolite in totality, while they can form segments of size three. For larger metabolites, the seven-atom segments are able to represent the structural similarity in a better way than the three or five-atom segments. Two structurally dissimilar molecules may have common three-atom segments, but the chance of having five-atom or seven-atom segment is comparatively less. In Section 6.4, it has been mathematically proved that a metabolite can be broken down into multiple 3-atom segments. Joining these three-atom segments will lead to the formation of the original atom. On the other hand, for five and seven-atom segments, one or more atoms may not find its place in any of the segments formed. Hence their amalgamation would not lead to the original 2D structure of the metabolite.

Let us consider p^{th} three-atom segment $X'^{(3)}_p = E_{i,k-1}E_{i,k}E_{i,k+1}$, q^{th} five-atom segment $X'^{(5)}_q = E_{i,k-2}E_{i,k-1}E_{i,k}E_{i,k+1}E_{i,k+2}$ and r^{th} seven-atom segment $X'^{(7)}_r = E_{i,k-3}E_{i,k-2}E_{i,k-1}E_{i,k}E_{i,k+1}E_{i,k+2}E_{i,k+3}$ of i^{th} metabolite, where $E_{i,k}$ is the kth atom of ith metabolite. Thus,

$$X_p^{(3)} = \{E_{i,k-1}, E_{i,k}, E_{i,k+1}\}; \tag{6.1}$$

$$X_q^{(5)} = \{E_{i,k-2}, E_{i,k-1}, E_{i,k}, E_{i,k+1}, E_{i,k+2}\}; \tag{6.2}$$

and

$$X_r^{(7)} = \{E_{i,k-3}, E_{i,k-2}, E_{i,k-1}, E_{i,k}, E_{i,k+1}, E_{i,k+2}, E_{i,k+3}\}; \tag{6.3}$$

where $1 \leq i \leq a$, and $E_{i,k}$ is not a terminal atom; $p, q, r \in \mathbb{N}$; p, q and $r = 1, 2, ...,$ such that

$$X_i''^{(3)} = \{X_p'^{(3)} | p \in \mathbb{N}\} \tag{6.4}$$

$$X_i''^{(5)} = \{X_q'^{(5)} | q \in \mathbb{N}\} \tag{6.5}$$

$$X_i''^{(7)} = \{X_r'^{(7)} | r \in \mathbb{N}\} \tag{6.6}$$

Two-dimensional structure of a metabolite can be depicted in the form of these segments. The segments can be combined to form a larger segment of any length. Initially two bonds with one common atom have been combined to form a three-atom segment. For example, bonds $E_{i,k-1}E_{i,k}$ and $E_{i,k}E_{i,k+1}$ have been combined together to form segment $E_{i,k-1}E_{i,k}E_{i,k+1}$, where $E_{i,k}$ is the common atom between the bond atoms. Subsequently, two three-atom segments having only one common terminal atom have been concatenated to form a five-atom segment. Likewise, a five-atom segment has been concatenated with a three-atom segment to form a seven-atom segment. For example, consider a certain three-atom segment $E_{i,k_1-1}E_{i,k_1}E_{i,k_1+1}$ and a certain five-atom segment $E_{i,k_2-2}E_{i,k_2-1}E_{i,k_2}E_{i,k_2+1}E_{i,k_2+2}$. If $k_1 - 1 = k_2 - 2$ or $k_1 - 1 = k_2 + 2$ or $k_1 + 1 = k_2 - 2$ or $k_1 + 1 = k_2 + 2$, these two segments can be concatenated to form a seven-atom segment.

The segments are formed following the rule such that all segments, except the first one should contribute to the addition of only one new atom. Consider a set F containing all the atoms $E_1, E_2, ...E_{B_i}$ of i^{th} metabolite whose segments need to be formed. Let $X_p'^{(3)} = E_{i,k-1}E_{i,k}E_{i,k+1}$ be the first continuous segment of length three. Hence, the corresponding $X_p^{(3)}$ is $\{E_{i,k-1}, E_{i,k}, E_{i,k+1}\}$. Initially, $X_i''^{(3)} = \phi$. Since $X_p'^{(3)}$ is the first segment formed, $X_i''^{(3)}$ is modified as $X_i''^{(3)} = X_i''^{(3)} \cup \{X_p'^{(3)}\}$. The atoms in $X_p^{(3)}$ are now removed from F. Hence, $F = F - X_p^{(3)}$. The second segment of length three has been formed in a way such that any one of the terminal atoms must be present in F while the other two atoms must not be present in F. Let the new segment formed be $X_p'^{(3)} = E_{i,k}E_{i,k+1}E_{i,k+2}$. Previously, the atoms $E_{i,k-1}, E_{i,k}$ and $E_{i,k+1}$ were removed from F. Comparing the previous and the

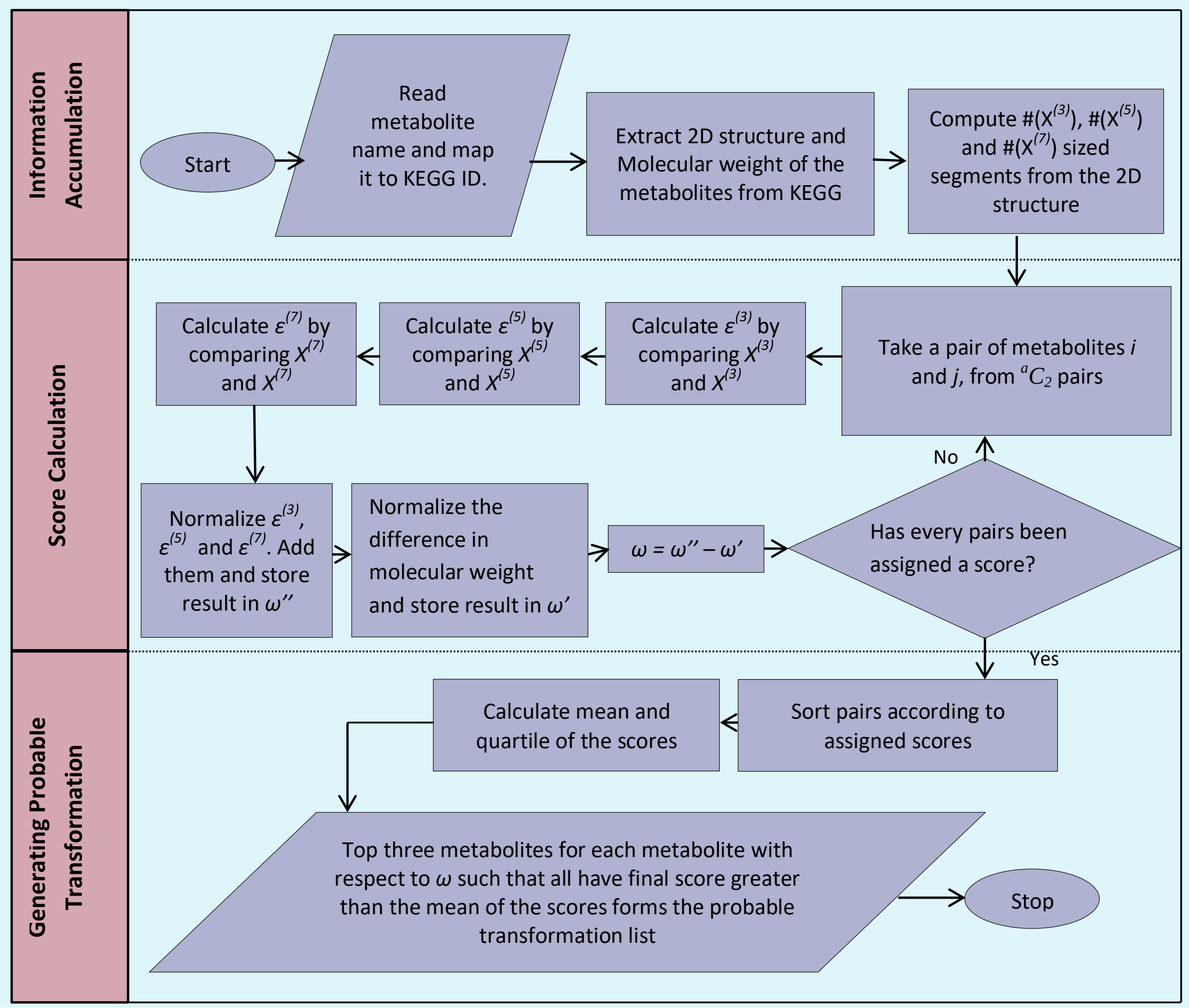

Information Accumulation
Score Calculation
Generating Probable Transformation
Start
Read metabolite name and map it to KEGG ID.
Extract 2D structure and Molecular weight of the metabolites from KEGG
Compute #($X^{(3)}$), #($X^{(5)}$) and #($X^{(7)}$) sized segments from the 2D structure
Take a pair of metabolites i and j, from ${}^{a}C_2$ pairs
Calculate $\varepsilon^{(3)}$ by comparing $X^{(3)}$ and $X^{(3)}$
Calculate $\varepsilon^{(5)}$ by comparing $X^{(5)}$ and $X^{(5)}$
Calculate $\varepsilon^{(7)}$ by comparing $X^{(7)}$ and $X^{(7)}$
Normalize $\varepsilon^{(3)}$, $\varepsilon^{(5)}$ and $\varepsilon^{(7)}$. Add them and store result in ω''
Normalize the difference in molecular weight and store result in ω'
$\omega = \omega'' - \omega'$
Has every pairs been assigned a score?
No
Yes
Sort pairs according to assigned scores
Calculate mean and quartile of the scores
Top three metabolites for each metabolite with respect to ω such that all have final score greater than the mean of the scores forms the probable transformation list
Stop

Figure 6.3: Flowchart of ASAPP

new segment formed, $E_{i,k}$ and $E_{i,k+1}$ are common atoms. These two atoms were already removed from F. The terminal atom $E_{i,k+2}$ is present in F. This atom, which is common in the new segment $X_p^{(3)}$ and F, has been removed from F. Thus, $X_i''^{(3)} = X_i''^{(3)} \cup X_p'^{(3)}$. Hence, for the segments, except the first one, we have

$$F = \begin{cases} F - (F \cap X_p^{(3)}), & \text{if } |F \cap X_p^{(3)}| = 1; \\ F, & \text{otherwise.} \end{cases} \tag{6.7}$$

Segments have been formed until F becomes empty, and only those segments have been retained, which have led to the removal of only one atom from F. Formation of the segments of size five and seven is a tweak of the above rule, such that, F may not be empty even after all possible unique segments are formed.

Consider Figure 6.2 for an example. There are $m = 10$ atoms in the metabolite, such that, $F = \{E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}\}$, where E_1=OH, E_2=C,... and so on. We aim at forming three-atom segments initially. The first segment $X_p'^{(3)} = E_1 - E_2 - E_3$ is formed such that $X_p^{(3)} = \{E_1, E_2, E_3\}$. Initially, $X_i''^{(3)} = \phi$. Since it is the first segment, $X_i''^{(3)} = X_i''^{(3)} \cup \{X_p'^{(3)}\}$. The atoms in the segments are removed from F. New F becomes $F = \{E_4, E_5, E_6, E_7, E_8, E_9, E_{10}\}$. Let the next segment formed be $X_p'^{(3)} = E_2 - E_3 - E_5$ such that $X_p^{(3)} = \{E_2, E_3, E_5\}$. According to the rule, E_5 is the only atom that is common in both $X_p^{(3)}$ and F, hence E_5 is removed from F, leading to $F = \{E_4, E_6, E_7, E_8, E_9, E_{10}\}$ and $X_i''^{(3)} = X_i''^{(3)} \cup \{X_p'^{(3)}\}$. Repeating the previous operation, the next segment formed is $X_p'^{(3)} = E_4 - E_3 - E_5$ such that $X_p^{(3)} = \{E_3, E_4, E_5\}$. According to the rule, E_4 is the only atom that is common in $X_p^{(3)}$ and F, hence E_4 is removed from F, leading to $F = \{E_6, E_7, E_8, E_9, E_{10}\}$ and $X_p^{(3)}$ is retained. Suppose the next segment formed is $E_2 - E_3 - E_4$ such that $X_p^{(3)} = \{E_2, E_3, E_4\}$. According to the rule, since $|F \cap X_p^{(3)}| \neq 1$, no deduction is performed in this step and $X_p^{(3)}$ is discarded. In this particular example, five-atom segments can be formed so that F becomes empty at the end. During the formation of seven-atom segment, the atom E_4 remains in F even after all the seven-atom segments have been formed. No seven-atom continuous segment containing the atom E_4 can be formed. The step of segmentation terminates when all possible three-atom, five-atom and seven-atom segments have been constructed. The next step is to find the similarity between pairs of metabolites in terms of common segment count.

6.2.3 Computing similarity between a pair of metabolites

Following the process of segmentation, the next step is to quantify the similarity between a pair of i^{th} and j^{th} metabolites. Considering a pair of metabolites, the number of common three-atom, five-atom and seven-atom segments between these are counted as follows:

$$\epsilon_{ij}^{(l)} = |X_i''^{(l)} \cap X_j''^{(l)}|, l = 3, 5, 7 \tag{6.8}$$

The score $\epsilon_{ij}'^{(l)}$, corresponding to $\epsilon_{ij}^{(l)}$ $(l = 3, 5, 7)$, has been obtained by standardizing the number of common segments as

$$\epsilon_{ij}'^{(l)} = \frac{\epsilon_{ij}^{(l)}}{|X_i''^{(l)} \cup X_j''^{(l)}| + \Delta_i + \Delta_j + a_i' + a_j'} \tag{6.9}$$

where Δ_i, Δ_j represents the number of bonds in the ith and jth metabolite and a_i', a_j' represents the number of atoms in the ith and jth metabolite.

Molecules of metabolites are of varying sizes. Hence, the count of common segments requires standardization. The scores have been standardized based on the complete structure of each metabolite. The similarity between two metabolites depends primarily on four factors:

1. Number of three-atom common segments between two metabolites.
2. Number of five-atom common segments between two metabolites.
3. Number of seven-atom common segments between two metabolites.
4. Difference in molecular weight of two metabolites.

The similarity score between a pair of metabolites has been found to increase with the number of common three-atom, five-atom and seven-atom segments. Higher the number of matched segments, higher is the structural similarity between a pair of metabolites. Factor 4 above has been found to have an inverse association with the similarity score. For most of the metabolites, it has been noticed that closer the two-dimensional structures of two metabolites, lower is the difference in the molecular weights. The standardized difference in molecular weights ω'_{ij} has been considered as a contributing factor for computing the similarity scores, and is defined as:

$$\omega'_{ij} = \frac{|w_i - w_j|}{\Delta_i + \Delta_j} \tag{6.10}$$

where w_i, w_j represents the molecular weights of the ith and jth metabolites. The summation of the individual scores $\epsilon'^{(3)}_{ij}, \epsilon'^{(5)}_{ij}, \epsilon'^{(7)}_{ij}$ for 3, 5 and 7 segments for each of the metabolites is given as ω''_{ij}. Thus, the final score ω_{ij} for each metabolite pair is

$$\omega_{ij} = \omega''_{ij} - \omega'_{ij} \tag{6.11}$$

6.2.4 Probable transformations

The metabolite pairs have been sorted in descending order of their final scores ω_{ij}, from highly probable to highly improbable transformation pairs. Mean, quartile and triplets have been used as the threshold values to isolate the probable transformations from the improbable ones. Using mean, the set of probable transformations are:

$$feasible\_pair = \left\{(\mathscr{C}_i, \mathscr{C}_j)|\omega_{ij} > \frac{\sum_{(i,j)=(1,1)}^{(a,a)} \omega_{ij}}{\binom{a}{2}}, i \neq j\right\} \tag{6.12}$$

The third quartile of the scores has been computed as another threshold value, where all the transformations having the value of ω greater than the third quartile (which is derived by considering the total scores, ω, of all the transformations and finding the third quartile of these scores), are predicted to be feasible. For each metabolite, three metabolites (other than the metabolite under consideration) have been filtered on the basis of similarity scores which has the maximum resemblance with the metabolite under consideration. The similarity score for i^{th} metabolite with the rest of the metabolites in the list have been sorted as:

$$\omega_{ij_1} \leq \omega_{ij_2} \leq \omega_{ij_3} \ldots \leq \omega_{ij_{a-1}}, i \neq j \tag{6.13}$$

Three metabolites having the highest similarity values with the i^{th} metabolite, are extracted as follows:

$$\mathscr{C}_i^{(1)} = \left\{\mathscr{C}_j | score(\mathscr{C}_j) = \omega_{ij_{a-1}}, i \neq j\right\} \tag{6.14}$$

$$\mathscr{C}_i^{(2)} = \left\{\mathscr{C}_j | score(\mathscr{C}_j) = \omega_{ij_{a-2}}, i \neq j\right\} \tag{6.15}$$

$$\mathscr{C}_i^{(3)} = \left\{\mathscr{C}_j | score(\mathscr{C}_j) = \omega_{ij_{a-3}}, i \neq j\right\} \tag{6.16}$$

Here $\mathscr{C}_i^{(1)}$ stands for the metabolite with maximum similarity to the i^{th} metabolite, $\mathscr{C}_i^{(2)}$ designates the metabolite with next best similarity with respect to $\mathscr{C}_i^{(1)}$, and $\mathscr{C}_i^{(3)}$ denotes the metabolite with the next to next best similarity with respect to $\mathscr{C}_i^{(1)}$. Due to the better performance of triplet method, the final list of transformations for i^{th} metabolite is $\mathscr{C}_i^{(1)}$, $\mathscr{C}_i^{(2)}$, and $\mathscr{C}_i^{(3)}$ respectively.

Algorithm 2 Architectural Similarity-based Automated Pathway Prediction (ASAPP)

```
Procedure $ASAPP$
Perform initialization
**while** $i \leq a$
    Compute all possible unique 3,5 and 7-atom segments and store
    them in $X_i^{\prime\prime(l)}$ where $l = 3, 5, 7$.
**while** $i \leq a$
    $j \leftarrow$ i + 1
    **while** $j \leq a$
        Compute the number of common sized segments in $X_i^{\prime\prime(l)}$
        and $X_j^{\prime\prime(l)}$ and store the value in $\epsilon_{ij}^{(l)}$.
Standardize the common segment count $\epsilon_{ij}^{(l)}$ as $\epsilon_{ij}^{\prime(l)}$.
Compute the segment score $\omega_{ij}^{\prime\prime}$ by summing $\epsilon_{ij}^{\prime(l)}$.
Calculate the effect of molecular weight $\omega_{ij}^{\prime}$ $\left(\frac{abs(w_i - w_j)}{\Delta_i + \Delta_j}\right)$
Generate the final score $\omega_{ij}$ by calculating $\left(\omega_{ij}^{\prime\prime} - \omega_{ij}^{\prime}\right)$
Sort $\omega_{ij}$ in descending order. Find the mean value of $\omega$
**while** $i \leq a$
    Prune $\omega$ for 3 metabolite with maximum similarity to $i$th metabolite
    Discard metabolites having $\omega_{ij}$ greater than the mean of $\omega$.
Output probable transformations
```

As a precautionary measure to ensure that unnecessary transformations are not reported, we have used the combined mean and the triplet parameters to generate the probable list of transformations. After the top three metabolites have been obtained based on the final similarity score ω_{ij}, these metabolites are filtered using the mean value $\Bigg(\sum\limits_{\substack{i,j=1\\i\neq j}}^{a} \omega_{ij}/\binom{a}{2}\Bigg)$. The metabolites which have scores greater than the mean value, are taken into consideration and the rest are discarded.

The overall complexity of ASAPP is $O(a^2a'^2)$, where a' is the maximum number of atoms in a metabolite and a is the total number of metabolites. The detailed complexity estimation has been given in next section. The mathematical validation of ASAPP has been given in Section 6.4. Algorithm 2 describes the step-wise computation of ASAPP.

6.3 Analysis of Time Complexity

For a metabolites, in the Information Accumulation module (Figure 6.3), the number of computations needed to acquire the structural information of each of the metabolites is a. In the Score Calculation module (Figure 6.3), considering each metabolite having a' atoms, the number of bonds is Δ'. Segmentation involves finding another bond having one common atom which leads to $\mathscr{O}(\Delta'^2)$ computations. Five-atom segments are formed by combining two three-atom segments. Since the maximum number of three-atom segments can be $a' - 2$ (Lemma 1), therefore the formation of five-atom segments results in a computation of $\mathscr{O}(a'^2)$. Seven-atom segments are formed by combining a five-atom segment and a three-atom segment. Since the maximum number of three-atom segments can be $a' - 2$ (Lemma 1) and the maximum number of five-atom segments can be $a' - 4$ (Lemma 2), therefore the computation time for forming seven atom segments is $\mathscr{O}(a'^2)$. Thus the total time for segmentation is $\mathscr{O}(\Delta'^2 + a'^2 + a'^2)$. Since in organic compounds, the number of bonds never exceeds the number of atoms, therefore the complexity of segmentation is $\mathscr{O}(a'^2)$. This operation has been done for a metabolites. Thus, the total computation required is $\mathscr{O}(aa'^2)$.

For finding similarity score between any two metabolites, the number of common three, five, and seven-atom segments need to be derived. In order to obtain the number of common three-atom segments, the set of three-atom segments of one metabolite is compared to the set of three-atom segments of another metabolite. Since the maximum number of three-atom segments is $a' - 2$, therefore, at most a'^2 operations are needed for the comparison. To find the number of common five-atom segment, the set of five-atom segments of one metabolite is compared to the set of five-atom segments of another metabolite. Since the number of five-atom segments will be limited to $a' - 4$ (Lemma 2), therefore, at most a'^2 operations are needed for the comparison. Similarly, to find the number of common seven-atom segments, the set of seven-atom segments of one metabolite is compared to the set of seven-atom segments of another metabolite. The maximum number of operations needed for this comparison is a'^2, since the number of seven-atom segments will be limited to $a' - 6$ (Lemma 6). Therefore, the total complexity of comparison is $\mathscr{O}(a'^2 + a'^2 + a'^2)$, which is equivalent to $\mathscr{O}(a'^2)$.

There are a total of $\frac{a(a-1)}{2}$ metabolite pairs, therefore the complexity for calculating the number common segments among all metabolite pairs is $\mathscr{O}(a^2a'^2)$. In order to obtain the difference in molecular weight, $\mathscr{O}(a^2)$ computations are required. Finding the resulting transformations by applying the multiple thresholds require a^2 operations. Hence the total complexity is $\mathscr{O}(aa'^2 + a^2a'^2 + a^2)$, which is equivalent $\mathscr{O}(a^2a'^2)$, such that a' is the maximum number of atoms in a metabolite among all the metabolites and a is the total number of metabolites.

6.4 Mathematical validation

As previously stated, two-dimensional structure of a metabolite can be depicted in the form of segments. The segments can be combined to form a larger segment of any length. In this section we provide some technical insights into segmentational aspect of ASAPP.

Lemma 1: Two-dimensional structure of a metabolite having τ atoms can be fully broken down into $\tau - 2$ three-atom segments, and *vice-versa*.

Proof: Here the number of atoms in the metabolite is τ.

Base case: When $\tau < 3$, the number of three-atom segment formed is zero.

Case 1: When $\tau = 3$, the number of three-atom segment formed is 1.

Case 2: When $\tau = 4$, the number of three-atom segments formed is 2.

Case 3: Consider $\tau = m$. Let there be a set κ containing the atoms of the metabolite. Let the metabolite have τ atoms, each atom being numbered uniquely. Let μ be a segment of length three, having three atoms and two bonds. The number of segment formed is 1 and the number of atoms becomes $\tau - 3$. The next segment μ of length three is formed such that it will have three atoms and either one of the terminal atoms must be present in κ while the other two atoms must not be present in κ. The atom that is common in the segments μ and κ, is removed from κ. The new count of segment is 2 and the new count of remaining atoms to form segments is $(\tau - 3) - 1$. For the 3rd segment formed, the count of atoms is $((\tau - 3) - 1) - 1$. The first segments results in deduction of three new atoms. After forming the first segment, for every new segment formed, one atom is being removed from τ. Total number of segments are $\tau - 3$. Hence there will be $1 + 1 * (\tau - 3)$ segments, which leads to $\tau - 2$ segment in total. The second part of this proof can be accomplished by bringing together all the three-atom segments formed in the first part, and combining the segments such that after each combination, either three or one new atom gets added to the set κ. Thus original metabolite can be formed from the segments.

Lemma 2: Two-dimensional structure of the metabolites having τ atoms can form at-most $\tau - 4$ such five-atom segments.

Proof: After the first segment is formed, the number of atoms is $\tau - 5$. On formation of the second segment, the number of atom becomes $(\tau - 5) - 1$. With the formation of the third segment, the number of atoms becomes $(\tau - 5) - 1 - 1$. The first segment results in removal of 5 atoms from τ. After forming the first segment, for every new segment formed, the total number of segments are $\tau - 5$. As previously mentioned, all the five-atom segments formed may not include all the atoms of the metabolite. Hence there will be at-most $1 + 1 * (\tau - 5)$ segments, which leads to $\tau - 4$ segments in total.

Lemma 3: Two-dimensional structure of the metabolites having τ atoms can form at-most $\tau - 6$ such seven-atom segments.

Proof: After the first segment is formed, the count of atom is $\tau - 7$. On formation of the second segment, the count of atom becomes $(\tau - 7) - 1$. With the formation of the third segment, the count of atom becomes $(\tau - 7) - 1 - 1$. The first segments results in removal of 7 atoms from τ. After forming the first segment, for every new segment formed, one atom is being deducted from τ. Total number of segments for which one atom is deducted from τ is $\tau - 7$. As previously mentioned, all the five-atom segments formed may not include all the atoms of the metabolite. Hence there will be $1 + 1 * (\tau - 7)$ segments at-most, which leads to $\tau - 6$ segment in total.

Lemma 4: If there is a set of A_X representing three-atom segments of a metabolite X and a set of $A_{X'}$ representing three-atom segments of another metabolite X', such that $A_X = B_{X'}$ (all the segments of A_X match all the segments of $A_{X'}$), then X = X'.

Proof: In connection to **Lemma 1**, all the three-atom segments in A_X can be combined together to form A. Same is the case with B. If the segments are the same, then the metabolite formed will also be the same.

6.5 Results

In this section, we shall describe how the algorithm has been applied to predict possible transformations in multiple crucial carbohydrates, lipid/fat and amino acid metabolic pathways. We have compared our results with the already established sets of transformations in KEGG.

6.5.1 Performance Comparison

ASAPP has been applied on 41 pathways involving 782 metabolites and 17556 transformation pairs as enlisted in KEGG. In order to analyze the performance of ASAPP, we have

Table 6.2: Performance comparison of various thresholding methods used in ASAPP

Performance measures	Mean	Quartile	Triplet
Accuracy	45.95	74.45	84.20
Sensitivity'	79.80	49.14	29.00
Specificity	43.14	75.77	86.13
F-score	59.36	63.28	61.74
G-mean	58.67	61.01	49.97

Table 6.3: Performance comparison (*accuracy*) of various threshold methods on the pathways of carbohydrate metabolism used in ASAPP. C* denotes the unique id of each pathways.

Pathway Name	Mean (%)	Quartile (%)	Triplet (%)
Glycolysis/ glycogenesis (C1)	45.56	75.86	83.25
TCA cycle (C2)	47.95	76.60	77.77
Pentose Phosphate Pathway (C3)	43.81	73.39	85.20
Pentose and Glucoronate inter-conversions (C4)	45.35	76.12	90.56
Galactose Metabolism (C5)	48.50	73.04	86.28
Pyruvate metabolism (C6)	70.68	74.63	63.79
Propanate Metabolism (C7)	53.04	75.41	89.14
Glyoxylate and dicarboxylate metabolism (C8)	53.11	74.77	91.10
Fructose and Mannose Metabolism (C9)	45.38	74.28	88.57
Starch and Sucrose Metabolism (C10)	50.36	74.20	87.10
Ascorbate and aldarate metabolism (C11)	39.03	75.30	89.36
Cs Branched dibasic acid metabolism (C12)	42.94	71.97	84.07
Inositol phosphate metabolism (C13)	50.33	76.92	86.23
Butanoate metabolism (C14)	45.38	76.02	87.56
Amino sugar and Nucleotide sugar metabolism (C15)	50.57	75.24	95.22

considered the transformations not enlisted in KEGG as not occurring at all. Such a consideration may not be correct since the presence of appropriate (yet unknown) enzymes may lead to the occurrence of such transformations. The summary of the performance measures has been depicted in Table 6.2.

A detailed description of the performance of the three categories of pathways (carbohydrate, lipid/fat and amino acid) have been depicted in the Tables 6.3, 6.4 and 6.5. Among the carbohydrate metabolic pathways, amino sugar and nucleotide sugar metabolism pathway has obtained the highest *accuracy* of 95.22%. Similarly, among the lipid pathways, the

Table 6.4: Performance comparison (*accuracy*) of various threshold methods on the pathways of lipid metabolism used in ASAPP. L* denotes the unique id of each pathways.

Pathway Name	Mean (%)	Quartile (%)	Triplet (%)
Alpha linoleic acid (L1)	52.43	77.80	88.41
Linoleic Acid metabolism (L2)	43.90	72.87	82.75
Arachidonic Acid Metabolism (L3)	36.72	74.12	91.37
Fatty Acid Elongation (L4)	21.33	76.00	83.33
Fatty acid Biosynthesis (L5)	43.66	76.87	90.47
Fatty acid degradation (L6)	68.90	76.97	89.07
Glycerophospholipid Metabolism (L7)	42.97	74.63	87.49
Glycerolipid Metabolism (L8)	47.81	75.86	84.13
Synthesis and degradation of ketone bodies (L9)	86.66	80.00	53.33
Sphinglipid Metabolism (L10)	47.07	74.55	87.94
Ether Lipid Metaboism (L11)	39.76	74.58	88.78
Primary Bile biosynthesis (L12)	35.84	75.28	93.98
Steroid Biosynthesis (L13)	43.00	74.00	77.33
Steroid Hormone Biosynthesis (L14)	46.93	75.65	84.49

Table 6.5: Performance comparison (*accuracy*) of various threshold methods on the pathways of aminoacid metabolism used in ASAPP. A* denotes the unique id of each pathways.

Pathway Name	Mean (%)	Quartile (%)	Triplet (%)
Alanine, aspartite and glumate metabolism (A1)	53.84	78.02	70.32
Valine, leucine and isoleucine biosynthesis (A2)	29.87	72.72	76.62
Lysine biosynthesis (A3)	41.12	74.89	78.78
Tryptophan metabolism (A4)	32.64	75.03	93.07
Valine, leucine and isoleucine degradation (A5)	50.58	74.28	86.89
Phenylalanine, tyrosine and tryptophan biosynthesis (A6)	47.05	73.20	85.21
Glysine, serine and threonine metabolism (A7)	47.95	74.08	88.70
Cysteine and methionine metabolism (A8)	48.74	74.24	90.72
Lysine degradation (A9)	43.81	75.24	87.19
Arginine Biosynthesis (A10)	42.85	62.63	60.43
Histidine Metabolism (A11)	43.29	74.65	89.54
Arginine and proline metabolism (A12)	46.47	74.24	92.27

primary bile biosynthesis pathway has achieved the highest *accuracy* of 93.98%. Finally, among the amino acid metabolism pathway, tryptophan metabolism has obtained the highest *accuracy* of 93.07%.

Considering all the pathways, a trade-off has been noticed among the *accuracy, sensitivity* and *specificity* (Table 6.2). When using the mean value of scores as a threshold, a high *sensitivity* but a low *accuracy* and *specificity* have been noticed, while on the other hand, the triplet method, a high *accuracy* and *specificity*, and a low *sensitivity* have been found. The quartile method has an average performance. Considering the three performance measures, we have chosen triplet method for prediction since it has given better performance in terms of accuracy and specificity, and has generated the least number of false positives.

Figure 6.4 shows the flow of synthesis and degradation of ketone bodies pathway formation using ASAPP involving 6 metabolites. The algorithm starts with a single compound. The initial metabolite considered here is a. High scores obtained by a is with the metabolites b ($\omega_{a,b}$=0.310) and c ($\omega_{a,c}$=0.301). Hence, we have obtained two new transformations from a, $a \rightarrow b$ and $a \rightarrow c$. Considering the newly obtained metabolite b, high score obtained is with a ($\omega_{a,b}$=0.310) and c ($\omega_{b,c}$=0.2888). Since a already exists in the pathway, the transition from b to c is added. Considering the newly obtained metabolite c, the high scores obtained are b ($\omega_{b,c}$=0.2888), a ($\omega_{a,c}$=0.301), and d ($\omega_{c,d}$=0.0960). Since a and b are already in the pathway, d is added to the existing pathway and a transition is made from c to d. With metabolite d, the high score obtained is with c ($\omega_{c,d}$=0.2960), e ($\omega_{d,e}$=0.1851) and f ($\omega_{d,f}$=0.2326). Metabolite c is already in the pathway, e and f are now added. Apart from the above mentioned pathway, the formation of six other pathways (alpha linoleic acid metabolism, linoleic acid metabolism, glycolysis pathway, TCA cycle, alanine aspartite and glutamate metabolism, and valine, leucine and isoleucine biosynthesis) have been depicted in Figures A.1 to A.6 in Appendix A. For a particular pathway, if the scores of most of the transformations are close to each other, then it can be concluded that the pathway constitutes structurally similar metabolites. Considering the alpha linoelic acid metabolism pathway (Figure A.1 in Appendix A) under the group of amino acid metabolism, it has been seen that apart from the transition between the molecule no. 24 (Traumatic acid) and 25 ((9Z,15Z)-(13S)-12,13-Epoxyoctadeca-9,11,15-trienoic acid), other transformations are associated with similar score among themselves, ranging from 0.265 to 0.294 (short interval) indicating that the compounds involved in this pathway are structurally similar to each other.

6.5.2 Application of ASAPP in the field of host-pathogen interactions

Toxins are substances secreted by plants and animals that are poisonous to humans. These toxins, once in the body of the host, intervene with the normal functioning of the metabolism of the host [195]. Pathogen liberated toxins have been seen to have a spectrum of upshots on

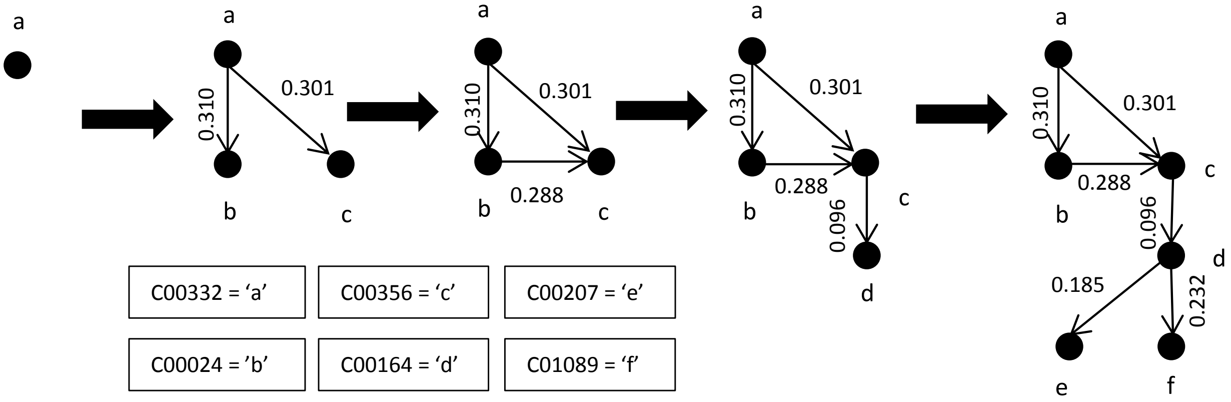


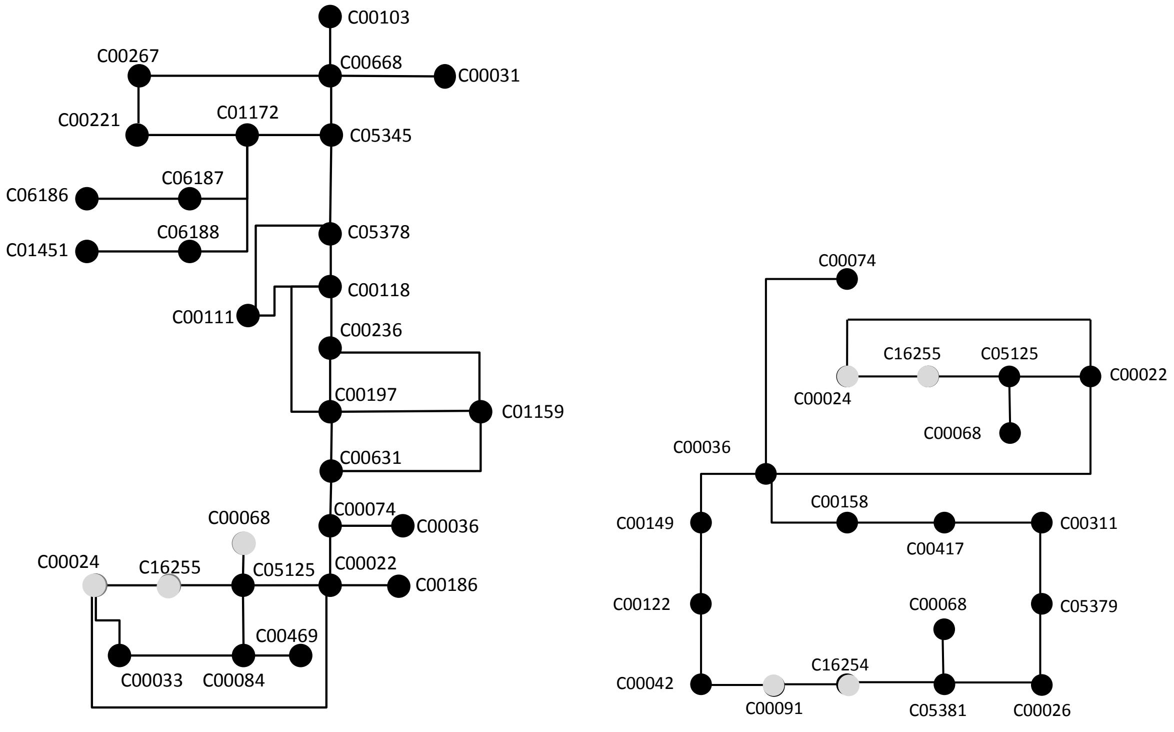
Figure 6.4: Step-by-step formation of the synthesis and degradation of ketone bodies pathway using ASAPP. a (Acetoacetyl-CoA), b (Acetyl-CoA), c (Hydroxymethylglutaryl-CoA), d (Acetoacetate), e (Acetone) and f ((R)-3-Hydroxybutanoate) are the compounds whose corresponding KEGG IDs are given. In each time step, one compound, whose transformations have not been considered previously and which is a recent addition to the pathway, is considered for finding the transformations related to that compound.

their hosts. The transformation mechanism of natural toxins need to be studied in details as these help in proper drug designing ([438]). The two-dimensional structural similarity of the toxins with the metabolites of metabolic pathways belonging to the host, are of significance and needs to be examined. Consider a simple pathway consisting of the transformations $A \rightarrow B, B \rightarrow$ C, and $C \rightarrow D$, where A, B, C, D are the compounds involved in the pathway. Consider a toxin X having high similarity with the metabolite B. Occurrence of an unknown reaction may block the transformation of $B \rightarrow C$. The other metabolites which react with B to produce C may as well, due to structural similarity and in the presence of appropriate enzyme, react with X to produce a different metabolite which is not C. Besides, if B is structurally similar to X, B can transform to X in the presence of appropriate enzyme and other metabolites. As soon as X is produced, the other metabolites, A, C, and D have a chance of reacting with X in the presence of the appropriate enzymes and thus breaking the pathway. The summary of the probable toxin transformations to/from metabolites from KEGG have been documented in the next section.

6.5.3 Effect of toxin on host

The adverse effect of toxins are as follows:

- Verruculogen (C20045), liberated by *Aspergillus* and *Penicillium* species, is carcinogenic, can weaken the immune system and is responsible for electro-physiological modifications of human nasal epithelial cells in vitro [215].
- Vindoline (C01626), secreted by *Vinca* species, is basically a fungal toxin, whose rapid



(a) Glycolysis

(b) TCA

Figure 6.5: The pathway models depicting the transformations within the (a) Glycolysis and (b) TCA pathway. The gray dots represent the breakpoints in the pathway. The black dots signify other metabolites which have a lower probability of being the breakpoints in the pathway.

spreading results in choking of native plant species and hence altering habitats. Adverse consequences of Vindoline include hair loss, loss of white blood cells and blood platelets, gastrointestinal problems, high blood pressure, excessive sweating, depression, muscle cramps, vertigo and headaches [257].

- Daturine (C02046) [45] and scopolamine (C01851), secreted by multiple species including *Anthocercis, Datura, Hyoscyamus* and *Mandragora*, causes poisoning. Symptoms of overdose include headache, nausea, vomiting, blurred vision, dilated pupils, hot dry skin, dizziness, dryness of the mouth, difficulty in swallowing, and central nervous system stimulation.

- Amygdalin (C08325) has genotoxic effect on cells and is cyanogenic in nature [296].

- Prunasin (C00844) emitted by *Sambucus* and *Pteridium* species, is a cyanogenic compound [169].

- 10-Deacetylbaccatin III (C11700) is disseminated by *Taxus* species and leads to headaches, lethargy, aching joints, itching, and skin rashes and in extreme cases, and it can have cancerous effect [153].

Table 6.6: Toxins having structural similarity with the metabolites of Glycolysis

KEGG Compound	KEGG Toxin
Thiamin diphosphate (C00068)	Brucine (C09084), Echimidine (C10299), Cylindrospermopsin (C19999), Gonyautoxin 1 (C16855), Philanthotoxin (C20052), Arenobufagin (C20035)
Acetyl-CoA (C00024)	alpha-Chaconine (C10796), Nodularin (C15713), Okadaic acid (C01945), Brevetoxin A (C16839), Azaspiracid (C16907)
S-acetyldihydrolipoyllysine (C16255)	alpha-Chaconine (C10796), Nodularin (C15713), Okadaic acid (C01945), Azaspiracid (C16907), Cephalostatin 1 (C20060)

6.5.4 Prediction of possible pathway breaks due to the presence of toxins

We have executed ASAPP on the metabolites involved in the Glycolysis and the TCA cycle. We have considered 52 toxins from KEGG. None of these toxins have any reported set of reactions in KEGG. For each of these toxins, we have predicted the consequence of its presence in the glycolysis (Figure 6.5 (a)) and the TCA cycle (Figure 6.5 (b)).

Considering Glycolysis pathway metabolite acetyl-coa (C00024), thiamin diphosphate (C00068) and s-acetyldihydrolipoyllysine (C16255) have the maximum chance of being the breakpoints of the pathway as depicted in Figure 6.5 (a) and Table 6.6. For example, toxin Anisatin (C09294) has high structural similarity with the metabolite beta-D-Fructose 1,6-bisphosphate (C05378). In presence of this toxin and appropriate enzyme, the metabolites that reacted with beta-D-Fructose 1,6-bisphosphate (C05378) to form D-Glyceraldehyde 3-phosphate (C00118) or Glycerone phosphate (C00111) may react with the toxin to produce unknown compounds in such a way that pathways is disrupted from its usual course. Among the rest of the metabolites, 1,3-bisphospho-d-glycerate (C00236), 2,3-bisphospho-d-glycerate (C01159), pyruvate (C00022), l-lactate (C00186), acetate (C00033), acetaldehyde (C00084), and ethanol (C00469) have been observed to have the least chance of being the breakpoints, i.e., the pathway has a high chance of not getting perturbed at these points. Considering the TCA cycle, the possible breakpoint metabolites are acetyl-coa (C00024), S-acetyldihydrolipoyllysine (C16255), succinyl-coa (C00091) and s-succinyldihydrolipoyllysine (C16254) as depicted in Figure 6.5 (b) and Table 6.7. Among the rest of the metabolites, pyruvate (C00022) and fumarate (C00122) have the least possibility of being the breakpoints. Further analysis leads to finding that the toxin azaspiracid (C16907) has the maximum likelihood to affect the Glycolysis and the TCA cycle as azaspiracid (C16907) has a high structural similarity with s-succinyldihydrolipoyllysine (C16254). Closely following

Table 6.7: Toxins having structural similarity with the metabolites in the TCA cycle

KEGG Compound	KEGG Toxin
Acetyl-CoA (C00024)	alpha-Chaconine (C10796), Nodularin (C15713), Okadaic acid (C01945), Brevetoxin A (C16839), Azaspiracid (C16907)
S-acetyldihydrolipoyllysine (C16255)	alpha-Chaconine (C10796), Nodularin (C15713), Okadaic acid (C01945), Azaspiracid (C16907), Cephalostatin 1 (C20060)
S-succinyldihydrolipoyllysine (C16254)	alpha-Chaconine (C10796), Pectenotoxin 1 (C16871), Brevetoxin A (C16839), Azaspiracid (C16907), Cephalostatin 1 (C20060)

Table 6.8: Toxin-based reactions found in Kegg. The 'Compound' column contains metabolites from the *Glycolysis* and *TCA* cycle. The column 'Type' denotes the type of reaction occurring between the toxin and the metabolite. 'Transformation' tag indicates the toxin and the metabolite are transformable to each other. 'Additive' tag indicates the toxin and the metabolite combine with each other to form a product.

Serial No.	Kegg Toxin	Kegg Compound	Type	Kegg Reaction ID
1	Verruculogen (C20045)	2-Oxoglutarate (C00026)	Transformation	R10445
2	Vindoline (C01626)	Acetyl-CoA (C00024)	Transformation	R03230
3	Daturine (C02046)	2-Oxoglutarate (C00026)	Additive	R03812
4	Scopolamine (C01851)	2-Oxoglutarate (C00026)	Transformation	R03737
5	Amygdalin (C08325)	D-Glucose (C00031)	Transformation	R02985
6	Prunasin (C00844)	D-Glucose (C00031)	Transformation	R02558
7	Prunasin (C00844)	D-Glucose (C00031)	Additive	R02985
8	10-Deacetylbaccatin III (C11700)	Acetyl-CoA (C00024)	Additive	R06311

these two toxins are the toxins nodularin (C15713) and okadaic acid (C01945), which too have a high chance of disrupting the pathways. A detailed result of the presence of toxins in the several pathways have been documented in the Table 6.8.

6.5.5 Analysis of ASAPP with respect to other algorithms

There exist several pathway prediction algorithms, viz., PathoLogic [212], PathMiner [272], Pathway hunter [324], Um-PPS [132], and Rahnuma [282]. However, ASAPP has a different aim compared to these methods. Besides, it has been noticed that apart from PathPred, none of the other algorithms are publicly available. Although PathPred is available, there is a fundamental difference between the functionality of PathPred and ASAPP. PathPred takes

Table 6.9: Comparative analysis of ASAPP with some existing algorithms

Tool name	Aim	Input	Output	Application domain	Web Availability
ASAPP	Predict possible pathway (linear/non-linear) among them	List of metabolites	Pathway with all the given metabolites predicted	All metabolic pathways	Available
PathoLogic [212]	Creating pathway genome database (PGDB) file	Annotated genome of an organism	PGDB file	Xenobiotic pathways	Unavailable
PathMiner [272]	Find linear path between these two compounds from KEGG	Initial metabolite, final metabolite in SMILES format	Linear pathway	Xenobiotic pathways	Unavailable
Pathway hunter [324]	Find shortest path between two metabolites using KEGG pathway information	Two metabolites in SMILES format	Shortest linear pathway	Xenobiotic pathways	Unavailable
PathPred [285]	Predict all pathways in which that metabolite is present from KEGG	One metabolite	Set of pathways	Xenobiotic pathways	Available
Um-PPS [132]	Recognize functional group in metabolite and apply group to group transformation as enlisted in UM-BBD database	One metabolite, draw the metabolite on MarvinView Java applet	Predict all pathways in which that metabolite is present	Xenobiotic pathways	Unavailable
Rahnuma [282]	Predict pathways using the metabolites from KEGG	KEGG pathways, metabolites	Pathways in which the metabolites occur	Bio-degradation pathways	Unavailable

in one metabolite as input and finds all the pathways involving that metabolite from KEGG database. PathPred does not predict a new pathway. ASAPP, on the other hand, takes a group of metabolites as inputs and predict possible pathways involving them. Moreover, unlike ASAPP, the functionality of PathPred is limited to xenobiotic pathways. Unavailability of the prediction algorithms and the limited functionality of PathPred makes ASAPP more significant. A summary of the analysis of ASAPP with respect to the other algorithms has been given in Table 6.9.

Prediction of host-pathogen interactions has been done at the population level [95], gene-level [329] and protein-level [12] [266] [297] [125]. At the population level, the statistics of the population of pathogen species interacting with host species are taken into consideration to predict novel interactions between a new pathogen and a host species [95]. At the gene-level, the pair of genes, one from the host and the other from the pathogen, is predicted to

Table 6.10: Analysis of prediction systems in the domain of host-pathogen interactions

Tool Description	Level	Aim
ASAPP	Metabolite	Predict possible pathway breaks due to toxins produced by pathogens
Dallas *et al.* [95]	Population	Predict connections between host species and pathogen species on a population level
Reid *et al.* [329]	Gene	Predict genes involved in host-pathogen interactions
Alguwaizani *et al.* [12]	Protein	Predict unknown PPI
Mariano *et al.* [266]	Protein	Predict unknown PPI
Nourani *et al.* [297]	Protein	Predict unknown PPI
Dyer *et al.* [125]	Protein	Predict unknown PPI

be interacting [329]. Host-pathogen interactions at the protein level are well studied. Host proteins, which interact with pathogen proteins, are predicted [266]. However, none has been done on the basis of metabolites and disruption of pathways. A summary of the analysis of ASAPP in the domain of host-pathogen interactions has been given in Table 6.10. ASAPP is one of a kind algorithm using which one can predict the probable pathway breaks in the host due to toxins from pathogens.

6.6 Conclusions

We have developed a novel algorithm ASAPP (Architectural Similarity-based Automated Pathway Prediction), which predicts biochemical transformations from the two-dimensional structure of metabolites. We have predicted the chance of transformation of one metabolite to another, depending on the two-dimensional structural similarity among the metabolites and the difference in their molecular weights. Based on these factors, we have given a score to each transformation and applied various threshold policies to determine the final list of probable transformations. Unlike other similar algorithms for pathway prediction, ASAPP has been made publicly available at `http://asapp.droppages.com/`.

By *in silico* analysis, we have shown how the presence of toxin in the host body may adversely affect its metabolic pathways. Here, we have predicted the outcome of 52 such toxins on the Glycolysis pathway and the TCA cycle. The effect of toxins on other pathways still needs to be explored. The field of host-pathogen interactions is emerging as a crucial area of infectious disease research in the post-genomic era. It is a budding research field where new discoveries are getting announced almost each day throughout the globe. The discovery of the dynamics of pathway perturbation during host-pathogen interactions will aptly facilitate further development in the field of discovering new drugs and new therapies for different diseases. Likewise, pathway perturbation is a crucial aspect of pathogen infection. Hence, further study on in this field is needed in the future.

Toxins not only affect metabolic pathways, but also affect signaling pathways. Thus, analyzing the effect of toxins on signaling pathways is a key issue in understanding host-pathogen interaction dynamics. In the next chapter, we analyze the effect of perturbation of signaling pathways by bacterial toxins. We develop a Boolean logic-based Network Robustness Analyzer (BNRA) that measures the robustness of a signal transduction pathway and analyzes the effect of perturbation of pathways due to toxins.

Chapter 7

Boolean Logic-based Network Robustness Analyzer (BNRA) and Its Application to a System of Host-Pathogen Interactions [352]

7.1 Introduction

The human body is made up of metabolic as well as signal transduction pathway. Signal transduction pathways regulate a wide spectrum of crucial cellular functions such as growth, differentiation, metabolism, and survival. Toxic proteins from pathogens can bind to the host proteins, which constitute such pathways, to alter key cellular functions or render them inactive. For example, tuberculosis causing pathogen *Mycobacterium tuberculosis* perturbs the PPM1A signaling pathway in macrophages. This action leads to the impairment of ability of macrophages to generate antimicrobial response [348]. The antimicrobial response pathway generates signals that help to defend microbial attack. Therefore, perturbation of such critical pathways that are the core to the immune response of the host, is life-threatening. Since signal transduction pathways play a crucial role in the human body, the study of the effect of toxins on pathways would be incomplete without the study of their effect on signal transduction pathways. In this chapter, we aim to analyze the effect of toxins on signal transduction pathways.

Signal transduction pathways are one of the most important biological networks in living cells. Perturbed signal transduction pathways result in many diseases, making it necessary to understand their mechanism. The availability of high-throughput data combined with the complexity of signaling mechanisms calls for a system-level understanding of signal transduction pathways. Two major computational approaches used to study signaling networks are graph theory and dynamical system modeling. Both the approaches are useful; network analysis (application of graph theory) helps us in understanding how the signaling network is organized and what its information-processing capabilities are, whereas dynamical modeling helps us in determining how the system changes with time and space upon receiving stimuli. Computational models have helped identify several emergent properties that signaling networks possess. Such properties include ultra-sensitivity, bi-stability, robustness, and noise-filtering capabilities. These properties equip cell-signaling networks with the capacity to disregard small or transient signals and/or amplify signals to drive cellular machines that spawn numerous physiological functions associated with different cell states. One of the crucial properties of signaling networks is robustness. Robustness of a network determines the ability of the network to preserve its dynamic behavior upon changes to its structure.

In the field of systems biology, representing signal transduction pathways in the form of Boolean networks are a very effective approach to computer-simulated analyses [213, 393]. A Boolean network is a directed graph where each vertex represents a protein in a biological network, and each edge between two vertices signifies the interaction between two proteins. Boolean networks have been used extensively, to trace the behavior of dynamic gene/protein networks. State transition matrices derived from the dynamic behavior of these systems allow application of standard inference methods to discover dependencies among the elements present in such a system [6].

Kauffman [213] and Thomas [393] have pioneered Boolean network modeling of signal transduction pathways. Akutsu *et al.* [5] and Dubrova *et al.* [121], have attempted to determine the stable states of a signal transduction pathway with the assumption that the in-degree for each gene/protein is less than or equal to two. Akutsu *et al.* have later extended this investigation to focus on the theoretical aspect of finding attractors [4]. Devloo *et al.* [112] have assumed the in-degree and out-degree of each gene/protein to be not more than three. However, in biological networks, in-degree/out-degree of genes/proteins cannot be restricted to a fixed number. In another investigation, Farrow *et al.* [138] have suggested a model that have derived the steady states via scalar equation approach using the sum of product form by utilizing the in-degrees of each of the genes/proteins. Yachie *et al.* [434] have described a method to determine the stable states of a network by specifying a set of interaction rules for a set of genes/proteins. However, in-applicability of these rules to all types of gene/protein pairs limits its versatility. Apart from that, their method has been restricted to pluripotent stem cells, hence not applicable to all signal transduction pathways, which makes the investigation purely domain-specific. Choo *et al.* [83] have developed an algorithm for identifying a particular phenotype of stable states for a large-scale Boolean network. However, the said algorithm could not derive the set of stable states for various networks, thus lacking versatility. Dubrova *et al.* have considered each node in the pathways to have at most two predecessors [121]. Such limitations restrict the scope of analysis of large pathways with highly connected hubs, i.e., vertices with many predecessors and successors.

Two significant drawbacks that govern the methods mentioned above are the size of the initial network and types of interactions considered. It has been noticed that current state-of-the-art investigations cannot process larger biological networks with more than 100 proteins. However, when a more extensive network of 100 or more vertices has been taken into consideration, the algorithms have failed to find the stable system states, robustness and stability of the whole system due to large memory and computational requirements. State-of-the-art algorithms have been able to handle only two types of interactions, activation, and inhibition [4,5,83,112,138,434], whereas, in the biological domain, many other interactions exist, like ubiquitination, dissociation, and binding among others. Every algorithm has a specified format for input/output. In this case, all the investigations have discussed the algorithm, but have not documented the type of input the algorithms work on and the format of output generated by them. The implementations of these algorithms have not been made available.

In this chapter, we have developed a Boolean logic-based Network Robustness Analyzer (BNRA) to determine the robustness of signaling networks. It ensures fast execution of the algorithm through the use of bit vectors, breaking the network into multiple subnetworks, and processing each subnetwork separately. The algorithm measures robustness of a network by generating valid states that the network can be in. BNRA also allows users to perturb the network, and visualize as well as quantify the change in its robustness due to perturbation. The chapter has been divided into multiple sections, starting with the methodology which describes the procedure of data collection. This is following by an elaborate description of the working mechanism of BNRA, along with the development of a scoring system for unperturbed networks and their perturbed counterparts. A step-by-step description of the application of BNRA on a sample pathway has been described in details. This is followed by the mathematical validation of BNRA, which outlines the mathematical properties of the algorithm. The derivation of time complexity of the algorithm has been furnished in the following section. BNRA has been applied to 221 pathways, which form the Results section of the chapter. Among the 221 pathways considered, BNRA has analyzed 73 disease pathways. Out of the 73 disease pathway considered, an analysis of nine of them have been provided. A comparative analysis of BNRA with state-of-the-art algorithms have been reported later.

7.2 Methodology

In this section, we develop the novel Boolean logic-based Network Robustness Analyzer (BNRA) for visualizing signaling networks as undirected graphs using which it analyzes the effect of perturbation on such network and calculates their robustness. The effect of perturbation is perceived by the change in robustness in a perturbed network from an unperturbed one. The flow of computation in BNRA has been depicted in Figure 7.1.

A biological signaling network has been considered as a graph G, such that $G = (V, E)$, V being the set of vertices representing proteins and E being the set of undirected edges representing the interaction type between pairs of proteins (vertices). These edges represent certain interactions between pairs of proteins leading to respective biological processes including regulation of genes. Regulation of gene expression includes a wide range of mechanisms that are used by cells to initiate or prevent the production of specific gene products (proteins or RNAs). A gene is said to be ON when it is activated to produce its specific gene product (protein) and is OFF when the corresponding protein is not produced [251].

The proposed algorithm BNRA assumes that the expression level of a protein can be one of two possible values (states), 1 (ON) and 0 (OFF). The presence of a protein is denoted by state 1 while its absence is denoted by state 0. The states of all the proteins in a network collectively depict a state of the network. A state of a network, involving the proteins (vertices), is represented by an n-dimensional vector [251]. The value of each subnetwork of the vector is 1 or 0 corresponding to the protein being ON/OFF. In the context of Boolean networks as models of signal transduction pathway, the binary approximation of gene expression is only, as Huang puts it [192], a “logical caricature”. However, although biological phenomena mostly manifest themselves in the continuous domain, they are often described in binary logical language as ON/OFF, up-regulated/down-regulated and responsive/non-responsive [361].

The edges in a graph represent the interactions between pairs of proteins. BNRA considers these edges to be undirected. Since an undirected graph has no concept of order, we have chosen to calculate the allowable states of all the proteins and interactions of the network simultaneously. This form of Boolean model is referred to as a synchronized Boolean model. The algorithm for BNRA has used the following terms:

- **Stable state:** A stable state of a network is referred to as a state of the network where all the interactions among the proteins are consistent [193]. Since each protein can have only one of the two states, there can be 2^n distinct states of a network of n proteins. Only a handful of these states form the stable-state. The stable states of a network often encode critical biological processes [193]. The stable state table for a sample network $(n = 7)$ has been given in Figure 7.2 (a). As observed, there are 8 stable states in T_1 $(r_{1,1}, r_{1,2}, \ldots, r_{1,8})$ for sample subnetwork G_1 and 3 stable states in T_2 $(r_{2,1}, r_{2,2}, r_{2,3})$ for sample subnetwork G_2 (Figure 7.2 (c)). Each of these stable states represents the stable states of the subnetwork, while the values 1/0 constituting each of these states represent the state of the 7 proteins. Higher number of stable states, *i.e.*, a large state space indicates that even if there is a perturbation in the protein state, the network has a higher chance of going into another stable state where the interactions among the proteins are consistent. On the other hand, in case of a lower number of stable states, perturbation may produce a set of states where the interactions among the proteins are not consistent. In other words, the chances of landing on to the states, where the interactions among the proteins are not consistent, are high. Hence, more the number of stable states, higher is the stability of the network.

- **Perturbation:** Here perturbation of a network is simulated by introducing noise for which state(s) of protein(s) change(s) [213]. One unit of noise, in this regard, changes the value of a single protein from 0 to 1 or vice versa. BNRA explores how such a single bit change affects the stability of the entire network.

- **Cycle:** From the stable states of a network, a new undirected graph $G' = (V', E')$ is formed. Each vertex v'_l in V'_l represents a stable state. Each edge e'_l in E'_l represents an undirected edge between two vertices $v'_{l,i}, v'_{l,i'}$, such that there is a Hamming distance of 1 between the two stable states $v'_{l,i}$ and $v'_{l,i'}$. Consider the sample network in Figure 7.2 (a). In Figure 7.2 (b), each of the stable states $r_{1,1}, r_{1,2}, \ldots, r_{1,8}$ correspond to vertices $v'_{1,1}, v'_{1,2}, \ldots, v'_{1,8}$ of G'. The vertices $v'_{1,1}$ $(r_{1,1} \equiv 0000)$ and $v'_{1,2}$ $(r_{1,2} \equiv 0100)$ will have an edge between them since they are 1-Hamming distance apart. However, the vertices $v'_{1,1}$ $(r_{1,1} \equiv 0000)$ and $v'_{1,3}$ $(r_{1,3} \equiv 1100)$ are not connected by an edge since they are 2-Hamming distance apart. In this way, we have got Figure 7.2 (e) where vertices (square shaped) represent stable states. A cycle in graph G' consists of three or more stable states such that any two states of the cycle differ exactly in a single position [213]. An example of a cycle in the sample network depicted in Figure 7.2 (e) is $v'_{1,3} - v'_{1,7} - v'_{1,6} - v'_{1,2}$. These cycles represent biological processes [318]. The cycles in a network's state space are called attractors [121]. By finding cycles, we aim at investigating the extent to which a network withstands perturbations without going into an unstable state. A detailed description of the working principle of BNRA on a sample pathway has been given in Section 2.3.

BNRA generates the stable state table and calculates robustness of signaling networks. It starts with the data collection phase in which it extracts information on signaling networks whose robustness is to be calculated. This is followed by filtering the interactions and fragmentation of the initial network. For each of the disconnected subnetworks obtained, the steps initialization, redundant copying, elimination, and robustness calculation are repeatedly performed. This is followed by computing $Rscore$, called robustness score for an unperturbed network, and $PRscore$, called robustness score for a perturbed network.

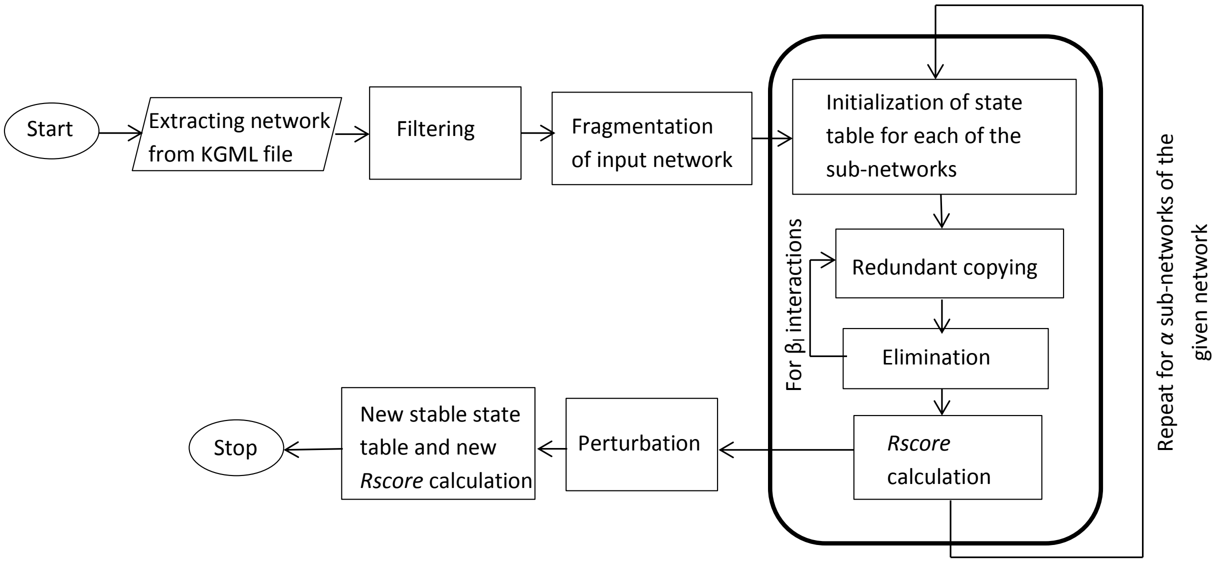

Start
Extracting network from KGML file
Filtering
Fragmentation of input network
Initialization of state table for each of the sub-networks
Redundant copying
Elimination
Rscore calculation
For βi interactions
Repeat for α sub-networks of the given network
Perturbation
New stable state table and new *Rscore* calculation
Stop

Figure 7.1: The diagram depicting the flow of the algorithm BNRA

7.2.1 Data Collection

Information on signaling networks has been obtained from Kyoto Encyclopedia of Genes and Genomes (KEGG) [209]. KEGG provides information about signaling networks in the form of the KEGG Markup Language (KGML) files (Section A.5.1 of Appendix A, Appendix B.4). BNRA considers KGML files as input and gives a measure of robustness of the network, before and after perturbation, as output.

7.2.2 Algorithm BNRA

Consider an undirected network $G(V, E)$ depicting proteins and interactions among them in a signaling network. The vertices represent proteins. An edge between a pair of vertices depicts the interaction between a pair of proteins. The usage of undirected edges is due to the fact that some of the interaction types cannot be represented by directed edges (Section 2.2.3). A summary of the variables used in the algorithm BNRA has been described in Table 7.1. The algorithm involves filtering, fragmentation, synchronized update, computation of robustness and exploring dynamics of altered behavior of the networks. We now describe each of these steps in detail. The flow of computation in BNRA is depicted in Figure 7.1.

Filtering

The interactions among pairs of proteins are of the format p [interaction type] q, where p and q are the proteins. These interactions are of the types *activation, inhibition, binding/association, ubiquitination, expression, phosphorylation, dephosphorylation, compound,*

Table 7.1: Summary of the variables used in BNRA

Name	Description
G	initial graph representing a given (initial) signaling network
V	set of vertices (proteins) in G
E	set of edges (interactions) in G
α	number of disconnected subgraphs (subnetworks) of G
G_l	lth subnetwork of $G, 1 \leq l \leq \alpha$
V_l	set of vertices in lth subnetwork of G
E_l	set of edges in lth subnetwork of G
n_l	number of proteins in lth subnetwork, i.e., $n_l = \|V_l\|$
β_l	number of interactions in lth subnetwork, i.e., $\beta_l = \|E_l\|$
p, q	proteins participating in an interaction
$e_{l,j}$	jth interaction of lth subnetwork G_l, where $1 \leq j \leq \beta_l$
$v_{l,j,1}$ and $v_{l,j,2}$	proteins involved in jth interaction of lth subnetwork G_l where $1 \leq j \leq \beta_l$
m_l	number of states of lth subnetwork
T_l	state table of lth subnetwork; $T_l = [t_{l,s,k}]$ where an entry $t_{l,s,k}$ is the state of kth protein in sth state from state table T_l of lth subnetwork, $1 \leq s \leq m_l, 1 \leq k \leq n_l$
$r_{l,s}$	sth ($1 \leq s \leq m_l$) state in state table T_l of lth subnetwork, i.e., sth row of T_l
$c_{l,k}$	states of kth ($1 \leq k \leq n_l$) protein in the state table T_l of lth subnetwork, i.e., kth column of T_l
S_l	ordered sequence of proteins forming state table T_l for lth subnetwork, $S_{l,k}$ denotes the protein corresponding to the kth column of T_l
M	table $M = [s', s'', l]$ where each row consists of 3 columns. An entry $< s', s'', l >$ denotes the row numbers s' and s'' $(s' \neq s'')$ representing two stable states $r_{l,s'}$ and $r_{l,s''}$ in T_l. Stable states $r_{l,s'}$ and $r_{l,s''}$ are 1-Hamming distance apart. The last column of M holds the corresponding subnetwork number l, $1 \leq l \leq \alpha$
G'_l	undirected graph formed using the entries in the first two columns of the table M
V'_l	set of vertices in G'_l, where each vertex v' corresponds to a row s' of a stable state $r_{l,s'}$ in T_l
E'_l	set of edges in G'_l, where each edge connects two vertices in G'_l representing two stables states that are 1-Hamming distance apart

missing and *indirect effect*. The interaction type *compound* gives intermediate of two interacting proteins, and cannot be represented in terms of 0/1. Hence, BNRA has filtered out these interactions from the initial set of interactions. The *missing* interactions have also been eliminated since information on these interactions are unavailable. Interactions of the type *indirect effect* too have been removed, since the effect between two participating proteins has no molecular details as to what effect one protein would have on the other when it is switched ON/OFF. BNRA has performed preprocessing steps to filter out these interactions from the initial set of interactions. The filtering module is followed by fragmentation module.

Fragmentation

In this module, we check if the initial (given) network can be partitioned into disconnected subnetworks. Most of the networks after getting filtered have been observed to be clustered into α disconnected subnetworks [213], such that

$$G(V, E) = G_1(V_1, E_1) \cup G_2(V_2, E_2) \cup \dots G_\alpha(V_\alpha, E_\alpha) \tag{7.1}$$

where $G_1, G_2, \dots G_\alpha$ represent α subnetworks of a network G, such that $V = \bigcup\limits_{l=1}^{\alpha} V_l, E = \bigcup\limits_{l=1}^{\alpha} E_l, V_l \bigcap\limits_{l \neq l'} V_{l'} = \varnothing$, $E_l \bigcap\limits_{l \neq l'} E_{l'} = \varnothing, 1 \leq l, l' \leq \alpha$, and $\varnothing$ being the null set. BNRA finds the disconnected subnetworks, if any, using Depth-First Search (DFS). For example, consider the sample network depicted in Figure 7.2 (a). The interaction *b missing e* is removed due to the filtering phase. This led to fragmentation of the initial network to two subnetwork, G_1 with vertices a, b, c, d and G_2 with vertices e, f, g.

Synchronized update: Generation of stable state table

In order to understand the stability of a network, a stable state table for each of the disconnected subnetworks has been generated. A stable state table is a collection of stable states (rows). Here, each row $r_{l,s}$ of the stable state table T_l depicts sth stable state of the stable state table T_l corresponding to lth subnetwork. Each column of T_l is denoted by $c_{l,k}$, and represents the states of kth protein for m_l stable states. Each row $r_{l,s}$ represents a stable state such that there are no conflicting interactions among the vertices denoted by the columns. For example, Figure 7.2 (b) is the stable state table for a sample network in Figure 7.2 (a). The number of stable states m_l for lth subnetwork having n_l proteins is always less than or equal to 2^{n_l}. BNRA uses a set of interaction rules, as described below, which are curated to form the stable states, and thereby stable state tables.

- *activation/expression*: The interaction 'p *activation/expression* q' indicates that protein p activates/expresses protein q [189]. For such an interaction, protein q can get activated/expressed in the presence of protein p. However, it may so happen that protein q is already activated/expressed by some other protein(s), even though protein p is OFF. Therefore, *activation/expression* is depicted by $00, 01, 11$ for p, q [6]. The situation that protein p is active but q is inactive after applying activation, is not possible and thus 10 is not feasible.

- *inhibition/repression*: The interaction 'p *inhibition/repression* q' indicates that protein p inhibits/deactivates protein q [189]. In such a case, when p is ON (active), q will be turned OFF (inactive), and if p is OFF, q will be turned ON. However, just like activation/expression, protein q can already be inhibited by some other protein(s), even though protein p is OFF. Therefore *inhibition/repression* is represented by $00, 01, 10$ for p, q [6], while 11 is not possible.
- *binding/association*: Two proteins p and q can bind if both of them are inactive, anyone of them is active or both are active. Therefore, in the present formulation, when two proteins bind, the corresponding *binding/association* is represented by $00, 01, 10, 11$ [295].
- *ubiquitination*: The process of ubiquitin getting attached to a protein sequence is *ubiquitination*. Ubiquitin exists in cells either freely or covalently conjugated with other proteins [400]. Since they are almost always present in a cell, unlike binding, if a protein is present in the cell, ubiquitination will occur. Otherwise, it will not occur. Hence, *ubiquitination* is represented by $00, 11$ [295].

Unlike *activation, inhibition, expression* and *repression*, other types of interactions, viz., *binding/association* and *ubiquitination*, cannot be given a direction. When proteins bind, the action of binding is not directed, *i.e.,* binding of protein p with q results in the same outcome as binding of protein q with p. Hence BNRA uses undirected networks to represent biological networks. Using the aforesaid rules, BNRA forms a stable state table for each of the subnetworks of the initial (given) network.

In order to construct the state table T, proteins and the interactions among them are taken into consideration. BNRA obtains the set of interactions from KEGG, which the algorithm processes to form the stable state table T. BNRA selects interactions one by one, and the possible effects of each of these interactions are reflected in the state table, and the proteins pertaining to this interaction are appended to S. Hence, $S_{l,k}$ contains the protein whose values corresponds to the entries in the kth column of T_l. The order of consideration of interactions to update the current state table does not affect the state of the final stable state table. Final stable state table is obtained after all the interactions of the network are considered, and the state table is updated accordingly. (Section 3, Lemma 1). However, the order of consideration of types of interactions has been observed to affect the execution time (Section 3, Lemma 3). In order to expedite processing, for each of the disconnected subnetworks, the interaction list is re-arranged such that the interactions *binding/association* are processed after all the other types of interactions are considered and their effect is incorporated in T (Section 3, Lemma 3).

Algorithm 3 Boolean logic-based Network Robustness Analyzer (BNRA)

```
for $l$ from $1$ to $\alpha$
    initialization()
    for $j$ from $1$ to $\beta_l$
        redundant_copying()
        elimination()
robustness_calculation()
```

Step 1: Initialization The execution of BNRA starts with the initialization phase. As explained earlier, BNRA considers a set of interactions from KEGG, which define the dynamics of the initial (given) network. In this step, the state table T for each of the disconnected subnetworks of the given network is initialized according to the first interaction encountered by BNRA. Consider lth subnetwork $G_l(V_l, E_l)$ of the initial network, where V_l is the set proteins and E_l is the set of interactions among them. The proteins associated with jth interaction $e_{l,j}$ in lth subnetwork are represented by $v_{l,j,1}$ and $v_{l,j,2}$. The state table T_l is a two-dimensional matrix which is initialized with the first interaction $e_{l,1}$ involving the proteins $v_{l,1,1}$ and $v_{l,1,2}$.

If the first interaction encountered by BNRA from the interaction set, corresponding to the lth subnetwork, is *inhibition/repression*, the state table T_l is initialized as

$$T_l = \begin{array}{c} \\ r_{l,1} \\ r_{l,2} \\ r_{l,3} \end{array} \begin{array}{c} \begin{array}{cc} c_{l,1} & c_{l,2} \end{array} \\ \left(\begin{array}{cc} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{array}\right) \end{array} \tag{7.2}$$

If the first interaction is *ubiquitination* then

$$T_l = \begin{array}{c} \\ r_{l,1} \\ r_{l,2} \end{array} \begin{array}{c} \begin{array}{cc} c_{l,1} & c_{l,2} \end{array} \\ \left(\begin{array}{cc} 0 & 0 \\ 1 & 1 \end{array}\right) \end{array} \tag{7.3}$$

If the first interaction is *activation/expression* then

$$T_l = \begin{array}{cc} & \begin{array}{cc} c_{l,1} & c_{l,2} \end{array} \\ \begin{array}{c} r_{l,1} \\ r_{l,2} \\ r_{l,3} \end{array} & \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \end{array} \tag{7.4}$$

If the interaction is *binding/association* then

$$T_l = \begin{array}{cc} & \begin{array}{cc} c_{l,1} & c_{l,2} \end{array} \\ \begin{array}{c} r_{l,1} \\ r_{l,2} \\ r_{l,3} \\ r_{l,4} \end{array} & \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} \end{array} \tag{7.5}$$

The size of T_l is depicted by m_l (number of rows) and n_l (number of columns), and sometimes by (m_l, n_l). For *binding/association*, the current size of T_l is $(4, 2)$, for *ubiquitination* the size is $(2, 2)$, and for other types of interactions, the sizes are $(3, 2)$. An entry $t_{l,s,k}$ in T_l denotes its element in sth row and kth column, such that $1 \leq s \leq m_l, \leq k \leq n_l$. A set S_l corresponding to the lth subnetwork is maintained, where S_l is the set of proteins

```
Initialization()
  if $e_{l,1}$ is activation/expression
    $t_{l,1,1} = 0$, $t_{l,1,2} = 0$, $t_{l,2,1} = 0$, $t_{l,2,2} = 1, t_{l,3,1} = 1$, $t_{l,3,2} = 1$
    $m_l = 3, n_l = 2$
  if $e_{l,1}$ is inhibition/repression
    $t_{l,1,1} = 0$, $t_{l,1,2} = 0$, $t_{l,2,1} = 0$, $t_{l,2,2} = 1, t_{l,3,1} = 1$, $t_{l,3,2} = 0$
    $m_l = 3, n_l = 2$
  if $e_{l,1}$ is ubiquitination
    $t_{l,1,1} = 0$, $t_{l,1,2} = 0$, $t_{l,2,1} = 1$, $t_{l,2,2} = 1$
    $m_l = 2, n_l = 2$
  if $e_{l,1}$ is binding/association
    $t_{l,1,1} = 0$, $t_{l,1,2} = 0$, $t_{l,2,1} = 0$, $t_{l,2,2} = 1$
    $t_{l,3,1} = 1$, $t_{l,3,2} = 0$, $t_{l,4,1} = 1$, $t_{l,4,2} = 1$
    $m_l = 4, n_l = 2$
    $S_l$ is initialized by appending it with $v_{l,1,1}$ and $v_{l,1,2}$
```

encountered by BNRA and already incorporated in T_l. When a new protein (not in S_l) is encountered, it is appended to S_l. Before initialization, S_l is empty. After initialization, proteins involved in the initial interaction are appended to S_l. S_l will eventually be of length n_l when all the interactions of lth subnetwork have been processed.

Step 2: Redundant copying A copy of the current state table is made and appended below current state table. After the initialization phase, state table T_l at any point of time may look like

$$T_l = \begin{array}{c} \\ r_{l,1} \\ r_{l,2} \\ \vdots \\ r_{l,m_l} \end{array} \begin{array}{c} \begin{array}{cccc} c_{l,1} & c_{l,2} & \cdots & c_{l,n_l} \end{array} \\ \begin{pmatrix} 1 & 1 & \dots & 1 \\ 0 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} \end{array} \tag{7.6}$$

In the next step, the next interaction from the list is processed. Let the upcoming interaction be $v_{l,j,1}$ *interaction* $v_{l,j,2}$, where $v_{l,j,1}, v_{l,j,2}$ are the proteins and an *interaction* can be one of *activation, inhibition, expression, repression, binding/association* or *ubiquitination*. In such a scenario, BNRA deals with three cases:

- Case 1: $v_{l;j,1} \notin S_l$ and $v_{l;j,2} \notin S_l$
- Case 2: $v_{l;j,1} \in S_l$ or $v_{l;j,2} \in S_l$
- Case 3: $v_{l;j,1} \in S_l$ and $v_{l;j,2} \in S_l$

where $1 \leq j \leq \beta_l$. For each of these cases, the copying mechanism varies.

Case 1 ($v_{l;j,1} \notin S_l$ and $v_{l;j,2} \notin S_l$): Both the proteins involved in the interaction are not in S_l. This indicates that the effect of the interaction needs to be reflected on T_l in the form of addition of two new columns $c_{n_{l+1}}, c_{n_{l+2}}$ to T_l. Proteins pertaining to the interaction is appended to S_l. The newly added columns of state table are populated with 0/1 accordingly. When the interaction is *binding/association*, the new state table T_l becomes

$$(T_l)_{4m_l,(n_l+2)} = \begin{array}{c} \\ r_{1:m_l} \\ r_{m_l+1:2m_l} \\ r_{2m_l+1:3m_l} \\ r_{3m_l+1:4m_l} \end{array} \begin{array}{c} \begin{array}{ccc} c_{1:n_l} & c_{n_{l+1}} & c_{n_{l+2}} \end{array} \\ \begin{pmatrix} (T_l)_{m_l,n_l} & 0 & 0 \\ (T_l)_{m_l,n_l} & 0 & 1 \\ (T_l)_{m_l,n_l} & 1 & 0 \\ (T_l)_{m_l,n_l} & 1 & 1 \end{pmatrix} \end{array} \tag{7.7}$$

Here, three copies of initial state table $(T_l)_{m_l,n_l}$ are made and placed below the initial set of states one after the other. The updated T_l is now of size $(4m_l, n_l + 2)$. Two proteins $v_{l,j,1}$ and $v_{l,j,2}$ have been put into S_l. S_l is of size $n_l + 2$. When the interaction is *activation* or *expression*, T_l becomes

$$(T_l)_{3m_l,(n_l+2)} = \begin{array}{cc} & \begin{array}{ccc} c_{l,1:n_l} & c_{l,n_{l+1}} & c_{l,n_{l+2}} \end{array} \\ \begin{array}{c} r_{1:m_l} \\ r_{m_l+1:2m_l} \\ r_{2m_l+1:3m_l} \end{array} & \begin{pmatrix} (T_l)_{m_l,n_l} & 0 & 0 \\ (T_l)_{m_l,n_l} & 0 & 1 \\ (T_l)_{m_l,n_l} & 1 & 1 \end{pmatrix} \end{array} \tag{7.8}$$

When the interaction is *inhibition* or *repression*, new T_l is

$$(T_l)_{3m_l,(n_l+2)} = \begin{array}{cc} & \begin{array}{ccc} c_{1:n_l} & c_{n_{l+1}} & c_{n_{l+2}} \end{array} \\ \begin{array}{c} r_{1:m_l} \\ r_{m_l+1:2m_l} \\ r_{2m_l+1:3m_l} \end{array} & \begin{pmatrix} (T_l)_{m_l,n_l} & 0 & 0 \\ (T_l)_{m_l,n_l} & 0 & 1 \\ (T_l)_{m_l,n_l} & 1 & 0 \end{pmatrix} \end{array} \tag{7.9}$$

Likewise, When the interaction is *ubiquitination*, T_l becomes

$$(T_l)_{2m_l,(n_l+2)} = \begin{array}{cc} & \begin{array}{ccc} c_{l,1:n_l} & c_{l,n_{l+1}} & c_{l,n_{l+2}} \end{array} \\ \begin{array}{c} r_{1:m_l} \\ r_{m_l+1:2m_l} \end{array} & \begin{pmatrix} (T_l)_{m_l,n_l} & 0 & 0 \\ (T_l)_{m_l,n_l} & 1 & 1 \end{pmatrix} \end{array} \tag{7.10}$$

The size of new T_l for *activation/expression/inhibition/repression* is $(3m_l, n_l + 2)$, for *ubiquitination* size is $(2m_l, n_l + 2)$, and for *binding/association* the size is $(4m_l, n_l + 2)$. S_l is updated by including $v_{l,j,1}$ and $v_{l,j,2}$ to its end. S_l is of size $n_l + 2$.

Case 2 **(**$v_{l;j,1} \in S_l$ **or** $v_{l;j,2} \in S_l$**):** In this case, one of the proteins, among the two proteins involved in the interaction being processed, is already in S_l. The other protein (not in S_l) needs to be added to update T_l, and is appended to S_l. For any of the interactions, the updated state table T_l becomes

$$(T_l)_{2m_l,(n_l+1)} = \begin{array}{ccc} & c_{l,1:n_l} & c_{l,n_{l+1}} \\ r_{1:m_l} & (T_l)_{m_l,n_l} & 0 \\ r_{m_l+1:2m_l} & (T_l)_{m_l,n_l} & 1 \end{array} \tag{7.11}$$

The updated T_l is now of size $(2m_l, n_l + 1)$. S_l is of size $n_l + 1$.

Case 3 ($v_{l;j,1} \in S_l$ and $v_{l;j,2} \in S_l$): In this case, both the proteins are already present in S_l and hence no change in S_l is needed. No change is needed for T_l in this case when the interaction is *binding/association* (Section 2, Lemma 2). For this interaction, elimination phase is skipped to process the next interaction since *binding/association* has no constraint, and all the previously processed interactions satisfy the interaction rules of the *binding/association* interaction. When the interaction is any of the other types, instead of *binding/association*, the algorithm directly goes over to the next step, i.e., *elimination*. Other types of interactions have constraints. For example, the *activation* interaction between two proteins does not allow the proteins to be in state 10, while *inhibition* does not allow the state of 11. Hence the elimination step (Step 3) needs to be executed over the current state table T_l to remove rows that do not satisfy the interaction rules. Consider the example depicted in Figure 7.3. In Step 5, the states r_5 and r_6 are removed since such states are not supported by *inhibition*. However, if an interaction b *binding/association* d, is encountered by BNRA such that both b and d are already present in S_l, then no states in T_l will be eliminated.

Step 3: Elimination Elimination, the final step of state table formation, follows the step of redundant copying (Step 2). In this step, the states (rows) which do not satisfy the interaction rules (Section 2.2.3) are removed. Consider an interaction $v_{l,j,1}$ *interaction* $v_{l,j,2}$, such that $1 \leq j \leq \beta_l$.

For Case 1 of Step 2, no elimination is needed. For Case 2 of Step 2, either $v_{l,j,1}$ or $v_{l,j,2}$ is already present in S_l. Let $S_{l,k}$ be the protein which is present in S_l, such that $1 \leq k \leq n_l$. For all the rows in T_l, the values in columns k and $n_l + 1$ must satisfy the interaction rules with respect to the interaction type between $v_{l,j,1}$ and $v_{l,j,2}$. If the current interaction is *activation*, *expression* or *ubiquitination*, the values in columns k and $n_l + 1$ should be $00, 01, 11$. Rows in which the entries in columns k and $n_l + 1$ are 1 and 0 respectively, have been removed. If the current interaction is *inhibition* or *repression*, the values in columns k and $n_l + 1$ should be $00, 01, 10$. Rows in which the values in columns k and n_{l+1} are 11 have been removed.

For Case 3, both $v_{l,j,1}$ and $v_{l,j,2}$ are present in S_l. The vertices $v_{l,j,1}$ and $v_{l,j,2}$ in S_l are

```
Redundant_copying()
  if ($e_{l,j}$ is binding/association) and ($v_{l,j,1} \notin S_l$ and $v_{l,j,2} \notin S_l$)
    Append $v_{l,j,1}, v_{l,j,2}$ to $S_l$
    $T'_l = T_l$
    Perform the operation [$T'_l$.append($T_l$)] 3 times
    $T_l = T'_l$
    $[m'_l, n'_l]$ =size($T'_l$)
    Initialize a new matrix $T_l$ of size $[m'_l, n'_l + 2]$
    $T_{l,1:m'_l,1:n'_l} = T'_{l,1:m'_l,1:n'_l}$
    $T_{l,1:m_l,n_l+1} = 0, T_{l,1:m_l,n_l+2} = 0$
    $T_{l,m_l+1:2m_l,n_l+1} = 0, T_{l,m_l+1:2m_l,n_l+2} = 1$
    $T_{l,2m_l+1:3m_l,n_l+1} = 1, T_{l,2m_l+1:3m_l,n_l+2} = 0$
    $T_{l,3m_l+1:4m_l,n_l+1} = 1, T_{l,3m_l+1:4m_l,n_l+2} = 1$
    $[m_l, n_l] = [m'_l, n'_l + 2]$
  if ($e_{l,j}$ is not binding/association) and ($v_{l,j,1} \notin S_l$ and $v_{l,j,2} \notin S_l$)
    Append $v_{l,j,1}, v_{l,j,2}$ to $S_l$
    $T'_l = T_l$
    if ($e_{l,j}$ is ubiquitination)
      $T'_l$.append($T_l$)
      $T_l = T'_l$
      $[m'_l, n'_l]$ =size($T'_l$)
      Initialize a new matrix $T_l$ of size $[m'_l, n'_l + 2]$
      $T_{l,1:m'_l,1:n'_l} = T'_{l,1:m'_l,1:n'_l}$
      $T_{l,1:m_l,n_l+1} = 0, T_{l,1:m_l,n_l+2} = 0$
      $T_{l,m_l+1:2m_l,n_l+1} = 1, T_{l,m_l+1:2m_l,n_l+2} = 1$
    else
      Perform the operation [$T'_l$.append($T_l$)] 2 times
      $T_l = T'_l$
      $[m'_l, n'_l]$ =size($T'_l$)
      Initialize a new matrix $T_l$ of size $[m'_l, n'_l + 2]$
      $T_{l,1:m'_l,1:n'_l} = T'_{l,1:m'_l,1:n'_l}$
      $T_{l,1:m_l,n_l+1} = 0, T_{l,1:m_l,n_l+2} = 0$
      $T_{l,m_l+1:2m_l,n_l+1} = 0, T_{l,m_l+1:2m_l,n_l+2} = 1$
      if ($e_{l,j}$ is inhibition/repression)
        $T_{l,2m_l+1:3m_l,n_l+1} = 1, T_{l,2m_l+1:3m_l,n_l+2} = 0$
      else
        $T_{l,2m_l+1:3m_l,n_l+1} = 1, T_{l,2m_l+1:3m_l,n_l+2} = 1$
    $[m_l, n_l] = [m'_l, n'_l + 2]$
```

```
if ($e_{l,j}$ is *binding/association*) and ($v_{l,j,1} \notin S_l$ or $v_{l,j,2} \notin S_l$)
    $T_l$.append($T_l$)
    $[m'_l, n'_l]$ =size($T_l$)
    $T_{l,1:m'_l,1:n'_l} = T'_{l,1:m'_l,1:n'_l}$
    Initialize a new matrix $T'_l$ of size $[m'_l, n'_l + 1]$
    $[m_l, n_l] = [m'_l, n'_l + 1]$
    $T'_{l,1:m_l,n_l+1} = 0, T'_{l,m_l+1:2m_l,n_l+1} = 1$
    $T = T'$
    if $v_{l,j,1} \in S_l$
        Append $v_{l,j,2}$ to $S_l$
    else
        Append $v_{l,j,1}$ to $S_l$
    if ($e_{l,j}$ is not *binding/association*)
        elimination()
if ($e_{l,j}$ is not *binding/association*) and ($v_{l,j,1}$ and $v_{l,j,2} \in S_l$)
    elimination()
```

represented by S_{l,k_1} and S_{l,k_2} such that $1 \leq k_1, k_2 \leq n_l, k_1 < k_2$. The indices of the vertices $v_{l,j,1}$ and $v_{l,j,2}$ in S_l are k_1 and k_2. For *inhibition/repression*, rows in T_l, where $(T_l)_{:,k_1} = 1$ and $(T_l)_{:,k_2} = 1$, are discarded. The term $(T_l)_{:,k}$ stands for the entries in all the rows of kth column. Similarly, for *activation/expression*, rows in T_l where $(T_l)_{:,k_1} = 1$ and $(T_l)_{:,k_2} = 0$ are discarded. For *ubiquitination*, rows in T_l where $(T_l)_{:,k_1} \neq (T_l)_{:,k_2}$ are discarded. No operation is needed when the interaction is *binding/association* since it allows all possible combinations of 0's and 1's (Section 3, Lemma 2). As soon as the elimination step is executed, the remaining interactions in lth subnetwork are taken into account, and Steps 2 and 3 are repeatedly performed until the interaction list gets exhausted.

Robustness

The robustness of a network determines its ability to withstand perturbation. BNRA determines the robustness of a network G based on two parameters - a robustness score termed as $Rscore$ introduced in this chapter, and the number of cycles (cycle count) obtained from the stable state table T_l (Section 2).

Initially, BNRA creates a table M. Each row $< s', s'', l >$ of M has 3 columns which holds the row numbers s' and s'' of the stable states which are 1-Hamming distance apart, and the subnetwork number l for which the stable states are considered. The total number of rows in M is the number of stable state pairs which are 1-Hamming distance apart. In order to populate M for the given network G, BNRA exhaustively finds the row numbers (s', s'') of all the pairs of stable states in T_l, which are 1-Hamming distance apart.

Using M, BNRA forms a graph G' from which number of cycles is to be determined.

```
Elimination()
  if $v_{l,j,1}$ or $v_{l,j,2} \in S_l$
    $k = j$ such that $S_{l,j} == v_{l,j,1}$ or $S_{l,j} == v_{l,j,2}$
    for $s$ from $1$ to $m_l$
      if $e_{l,j}$ is association/expression and $t_{l,s,k} == 1$ and $t_{l,s,n_l+1} == 0$
        Delete $T_{l,s}$
        $m_l = m_l - 1$
      if $e_{l,j}$ is inhibition/repression and $t_{l,s,k} == 1$ and $t_{l,s,n_l+1} == 1$
        Delete $T_{l,s}$
        $m_l = m_l - 1$
      if $e_{l,j}$ is ubiquitination and $t_{l,s,k} \neq t_{l,s,n_l+1}$
        Delete $T_{l,s}$
        $m_l = m_l - 1$
  if $v_{l,j,1} \in S_l$ and $v_{l,j,2} \in S_l$
    $k_1 = j$ such that $S_{l,j} == v_{l,j,1}$, $k_2 = j$ such that $S_{l,j} == v_{l,j,2}$
    for $s$ from $1$ to $m_l$
      if $e_{l,j}$ is activation/expression and $t_{l,s,k_1} == 1$ and $t_{l,s,k_2} == 0$
        Delete $T_{l,s}$
        $m_l = m_l - 1$
      if $e_{l,j}$ is inhibition/repression and $t_{l,s,k_1} == 1$ and $t_{l,s,k_2} == 1$
        Delete $T_{l,s}$
        $m_l = m_l - 1$
      if $e_{l,j}$ is ubiquitination and $t_{l,s,k_1} \neq t_{l,s,k_2}$
        Delete $T_{l,s}$
        $m_l = m_l - 1$
```

Like G, G' is a collection of G'_l such that G'_l is formed based on entries in rows for which values in 3rd column of M are l. The first two elements of each tuple in M, i.e., s' and s'', are two vertices connected by an edge in G'. Thus each vertex pair (s', s'') represents an edge between them. The number of cycles of length ranging from 3 to 20 has been found from G' by using DFS algorithm. It may be mentioned here that cycles of length 2 cannot be formed. BNRA has found cycles of maximum length 20 for the 221 pathways considered here. Consider the example, for which the table M and the graph G' of the sample network given in Figure 7.2 (a) are depicted in Figure 7.2 (c) and Figure 7.2 (d) respectively.

A cycle is made up of multiple stable states, where one stable state differs by one bit from the stable states which are its immediate neighbors. The network/subnetwork can exist in one of the stable states. It is now perturbed by introducing one unit of noise through arbitrarily changing the value of a single protein. After perturbation, the system may return to one of the stable states in the cycle, move to a stable state belonging to a different cycle, or may land onto an unstable state. It may be mentioned here that these cycles represent biological processes [318]. The stages of a biological process are represented by these stable states in the cycles. Hence, more the number of cycles, higher is the robustness of the network [213].

We now define $Rscore$ as the average of the robustness scores of the individual subnetworks in a network. $Rscore$ reflects the stability of the entire network. Robustness score of each subnetwork is the ratio of the total number of stable states m_l obtained, to the possible number of states for n_l proteins. Thus, $Rscore$ of G is defined as

$$Rscore = \frac{1}{\alpha} \sum_{l=1}^{\alpha} \frac{m_l}{2^{n_l}} \tag{7.12}$$

The value of $Rscore$ lies between 0 and 1. Higher the number of stable states, higher is the value of $Rscore$. Higher number of stable states indicates a lower number of inconsistent states. Therefore, higher the value of $Rscore$, more robust the network is.

Perturbation

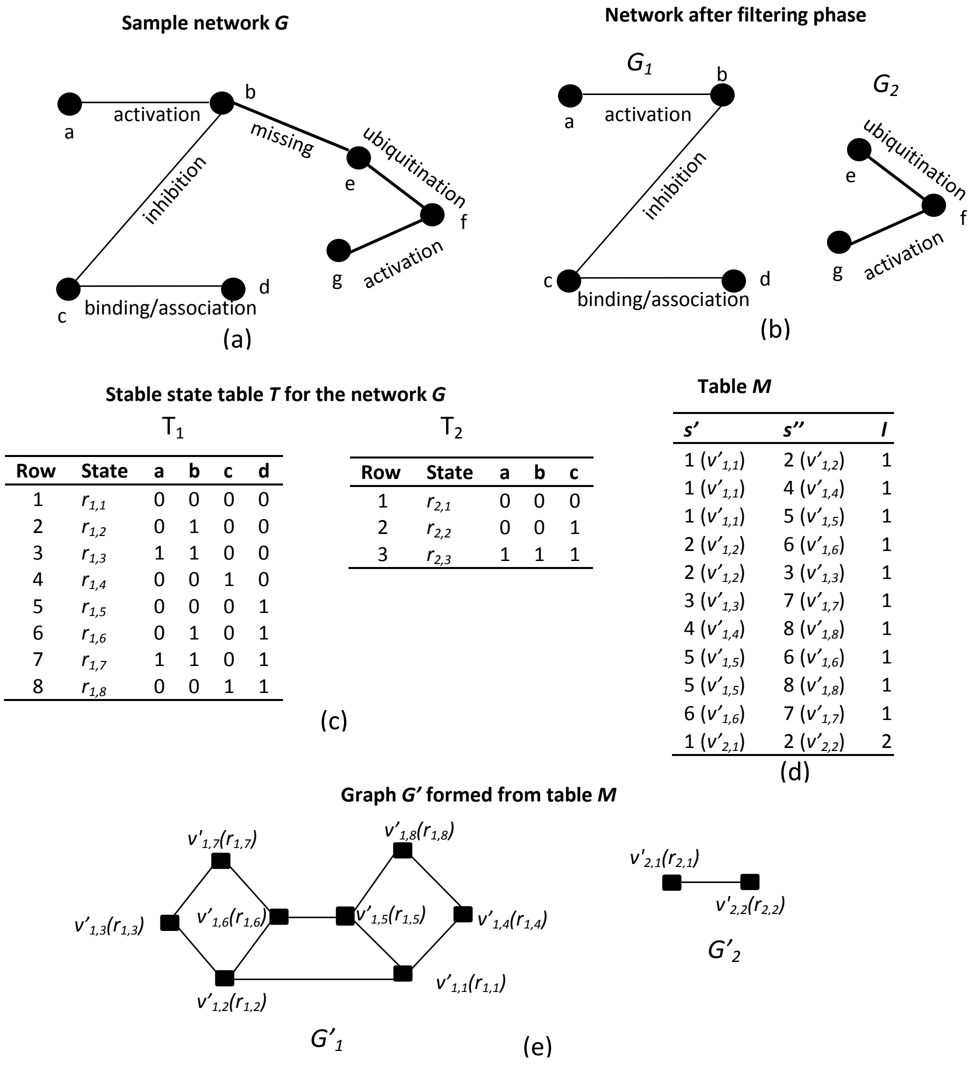
When a network is subjected to perturbation, the state of one or multiple proteins within the network changes [22]. This may affect the normal functioning of the pathway and its stability [213]. The network's tolerance towards perturbation determines the robustness of a network. BNRA finds the change in robustness of a network when an external protein (toxin) has an effect (i.e., *inhibition, activation* and so on) on one or more proteins of the network. The robustness of the new network after introduction of a new toxin has been derived in the form of $PRscore$, where

$$PRscore = \frac{1}{\alpha} \sum_{l=1}^{\alpha} \frac{m_l''}{2^{n_l''}} \tag{7.13}$$

The term m_l'' is the number of stable states and $n_l''(= n_l + 1)$ is the number of proteins after a toxin is introduced into the lth component of the network. It has been observed that if highly connected proteins are perturbed, the stability of a network is affected by a greater extent than perturbing other proteins.

BNRA is equipped to deal with the issue of perturbation, which it solves without recomputing the stable state table. When an additional input is given in the form of a new protein and its interaction with an existing protein belonging to the initial network, BNRA efficiently computes the modified $Rscore$ along with the cycle count. From the perturbed network, the $PRscore$ is calculated as above.

Before perturbation, the stable state table is denoted by T_l. On perturbation by a toxin, say t, the stable state table $(T_l)_{m_l,n_l}$, is altered by adding a new column for toxin t as shown in Figure 7.4. The step of redundant copying and elimination is carried out for t, to remove rows with inconsistent entries. After elimination, the modified number of stable states is denoted by m_l''. The modified number of nodes n_l'' after perturbation is $n_l + 1$. Thus, the size of the stable state table T_l becomes (m_l'', n_l''). A detailed description of perturbation with an example has been furnished in Section 2.3.



T_1

Row	State	a	b	c	d
1	$r_{1,1}$	0	0	0	0
2	$r_{1,2}$	0	1	0	0
3	$r_{1,3}$	1	1	0	0
4	$r_{1,4}$	0	0	1	0
5	$r_{1,5}$	0	0	0	1
6	$r_{1,6}$	0	1	0	1
7	$r_{1,7}$	1	1	0	1
8	$r_{1,8}$	0	0	1	1

T_2

Row	State	a	b	c
1	$r_{2,1}$	0	0	0
2	$r_{2,2}$	0	0	1
3	$r_{2,3}$	1	1	1

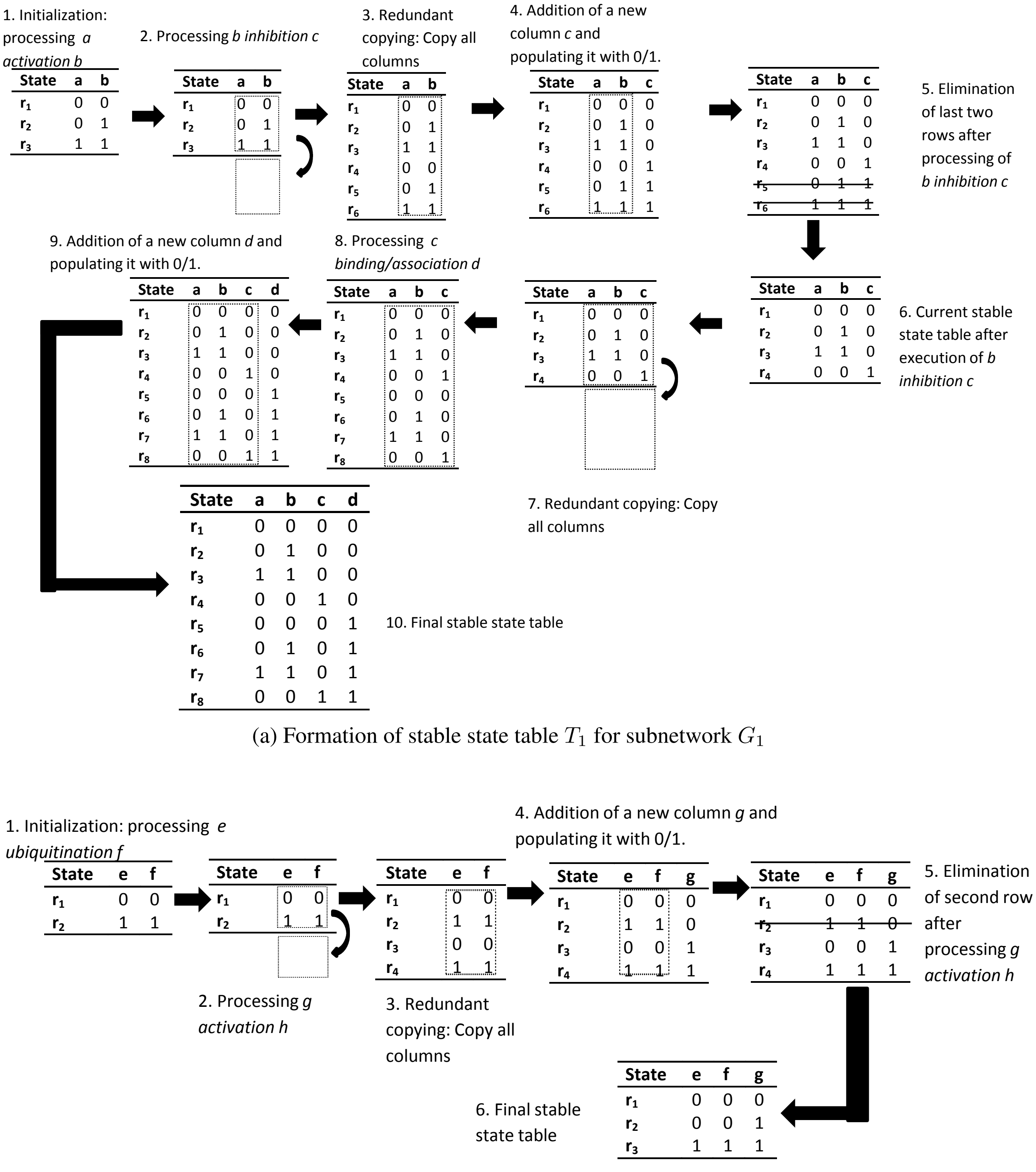
Table *M*

s′	*s″*	*l*
1 ($v'_{1,1}$)	2 ($v'_{1,2}$)	1
1 ($v'_{1,1}$)	4 ($v'_{1,4}$)	1
1 ($v'_{1,1}$)	5 ($v'_{1,5}$)	1
2 ($v'_{1,2}$)	6 ($v'_{1,6}$)	1
2 ($v'_{1,2}$)	3 ($v'_{1,3}$)	1
3 ($v'_{1,3}$)	7 ($v'_{1,7}$)	1
4 ($v'_{1,4}$)	8 ($v'_{1,8}$)	1
5 ($v'_{1,5}$)	6 ($v'_{1,6}$)	1
5 ($v'_{1,5}$)	8 ($v'_{1,8}$)	1
6 ($v'_{1,6}$)	7 ($v'_{1,7}$)	1
1 ($v'_{2,1}$)	2 ($v'_{2,2}$)	2

Figure 7.2: The diagram depicting the flow of the algorithm BNRA with an example. Figure 7.2 (a) depicts the initial network G; Figure 7.2 (b) shows the initial network G being fragmented into subnetworks G_1 and G_2. The stable state tables T_1 and T_2 for the subnetworks G_1 and G_2 respectively are shown in Figure 7.2 (c). Figure 7.2 (d) depicts table M formed from T_1 and T_2. Figure 7.2 (e) provides the graphs G_1' and G_2' obtained from M.

7.2.3 Execution of BNRA on a sample pathway

Consider a sample network as given in Figure 7.2 (a). It involves seven proteins, *viz.*, a, b, c, d, e, f, g, and six interactions. Each row in the table of Figure 7.2 (c) represents the list of stable states, *viz.*, $r_{1,1}, r_{1,2}, \ldots, r_{1,8}$ of subnetwork G_1 and $r_{2,1}, r_{2,2}, r_{2,3}$ of subnetwork G_2. The steps for obtaining the final stable state tables for each of the subnetworks has been depicted in Figure 7.3.



1. Initialization: processing *a activation b*

State	a	b
r_1	0	0
r_2	0	1
r_3	1	1

2. Processing *b inhibition c*

State	a	b
r_1	0	0
r_2	0	1
r_3	1	1

3. Redundant copying: Copy all columns

State	a	b
r_1	0	0
r_2	0	1
r_3	1	1
r_4	0	0
r_5	0	1
r_6	1	1

4. Addition of a new column *c* and populating it with 0/1.

State	a	b	c
r_1	0	0	0
r_2	0	1	0
r_3	1	1	0
r_4	0	0	1
r_5	0	1	1
r_6	1	1	1

State	a	b	c
r_1	0	0	0
r_2	0	1	0
r_3	1	1	0
r_4	0	0	1
~~r_5~~	~~0~~	~~1~~	~~1~~
~~r_6~~	~~1~~	~~1~~	~~1~~

5. Elimination of last two rows after processing of *b inhibition c*

State	a	b	c
r_1	0	0	0
r_2	0	1	0
r_3	1	1	0
r_4	0	0	1

6. Current stable state table after execution of *b inhibition c*

State	a	b	c
r_1	0	0	0
r_2	0	1	0
r_3	1	1	0
r_4	0	0	1

7. Redundant copying: Copy all columns

8. Processing *c binding/association d*

State	a	b	c
r_1	0	0	0
r_2	0	1	0
r_3	1	1	0
r_4	0	0	1
r_5	0	0	0
r_6	0	1	0
r_7	1	1	0
r_8	0	0	1

9. Addition of a new column *d* and populating it with 0/1.

State	a	b	c	d
r_1	0	0	0	0
r_2	0	1	0	0
r_3	1	1	0	0
r_4	0	0	1	0
r_5	0	0	0	1
r_6	0	1	0	1
r_7	1	1	0	1
r_8	0	0	1	1

State	a	b	c	d
r_1	0	0	0	0
r_2	0	1	0	0
r_3	1	1	0	0
r_4	0	0	1	0
r_5	0	0	0	1
r_6	0	1	0	1
r_7	1	1	0	1
r_8	0	0	1	1

10. Final stable state table

(a) Formation of stable state table T_1 for subnetwork G_1

1. Initialization: processing *e ubiquitination f*

State	e	f
r_1	0	0
r_2	1	1

2. Processing *g activation h*

State	e	f
r_1	0	0
r_2	1	1

3. Redundant copying: Copy all columns

State	e	f
r_1	0	0
r_2	1	1
r_3	0	0
r_4	1	1

4. Addition of a new column *g* and populating it with 0/1.

State	e	f	g
r_1	0	0	0
r_2	1	1	0
r_3	0	0	1
r_4	1	1	1

5. Elimination of second row after processing *g activation h*

State	e	f	g
r_1	0	0	0
~~r_2~~	~~1~~	~~1~~	~~0~~
r_3	0	0	1
r_4	1	1	1

6. Final stable state table

State	e	f	g
r_1	0	0	0
r_2	0	0	1
r_3	1	1	1

(b) Formation of stable state table T_2 for subnetwork G_2

Figure 7.3: Diagram depicting steps for the formation of the final stable state table by BNRA from the initial sample network given in Figure 7.2 (a).

We start with the initial network G given in Figure 7.2 (a). In the filtering step, the interaction b *missing* e is removed. In the fragmentation step, the initial network G is fragmented into two subnetworks, G_1 and G_2 as shown in Figure 7.2 (b). The stable state table for each of the subnetworks can be formed in any order (Section 3, Lemma 1). However, for simplicity, with regards to this example, formation of the stable state table of the subnetwork G_1 is carried out first followed by that of the other subnetwork G_2. The steps for the formation of stable state table T_1 for the subnetwork G_1 is depicted in Figure 7.3 (a), while the formation of T_2 for subnetwork G_2 is shown in Figure 7.3 (b). In the first subnetwork G_1, the first interaction to be processed is a *activation* b. It starts with initialization of a state table as shown in Figure 7.3 (a). S_1 has been updated from null to having elements a, b. T_1 is currently of size $(3, 2)$ as depicted in Step 1 of Figure 7.3 (a).

The next interaction to be processed is b *inhibition* c as depicted in Step 2 of Figure 7.3 (a). One of the proteins, i.e., b is already in S_1. Hence, a new column needs to be added to T_1 to include protein c. Before a new column is added, T_1 has been made to go through the step of redundant copying. Therefore, a single copy of the current state table T_1 has been appended to the end of the current T_1, as depicted in Step 3 of Figure 7.3 (a), which results in T_1 having the size of $(6, 2)$. Following this, a new column is added. Protein c has now been added to T_1 in the form of the new column, and populated with 0's and 1's as given in Step 4 of Figure 7.3 (a). Current size of T_1 is $(6, 3)$. S_1 has been updated to have the elements a, b, c. Since the interaction type is *inhibition*, states r_5, r_6 (rows 5 and 6 of Step 5) do not satisfy the interaction rules, have thus been eliminated from T_1. Therefore, the current size of T_1 is $(4, 3)$ (Step 6).

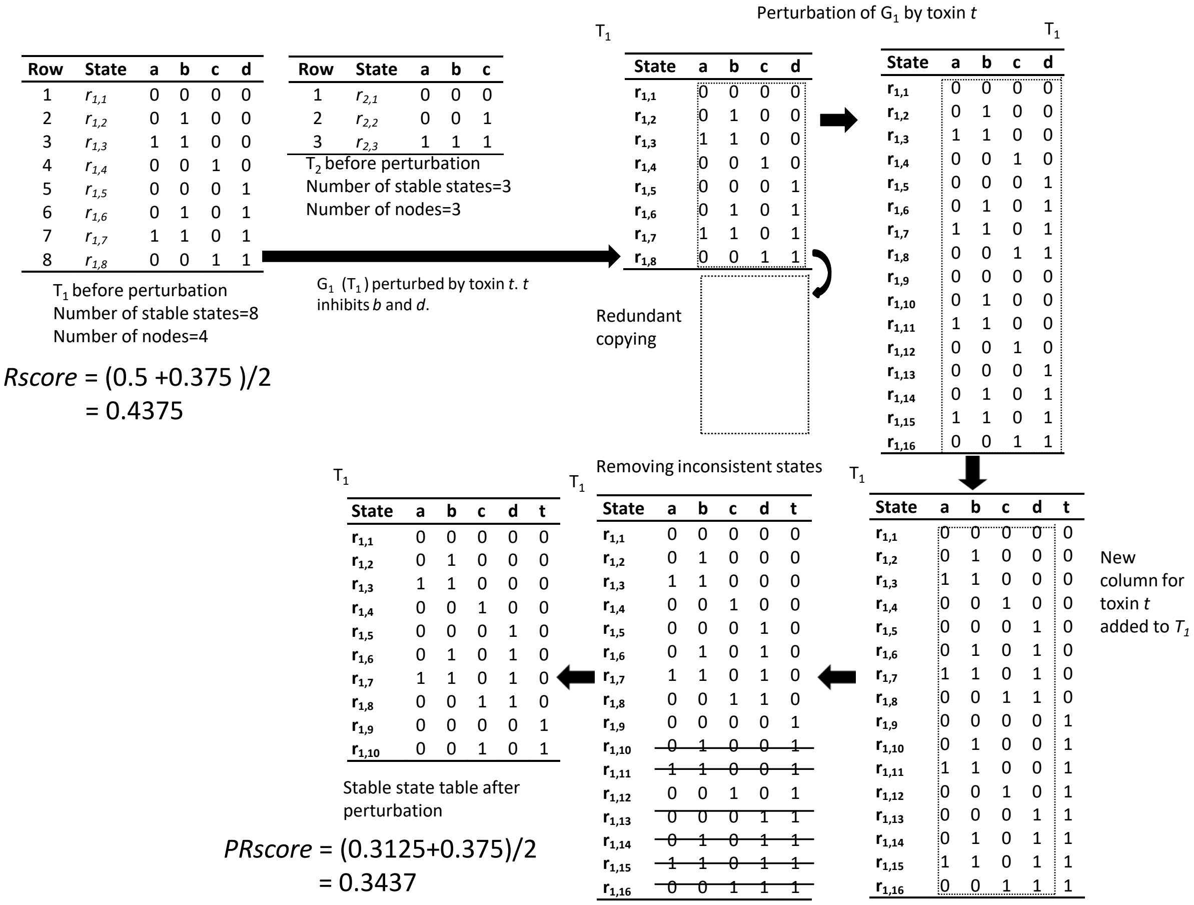
The next interaction to be processed is c *binding/association* d. One of the proteins, i.e., c is already in S_1. Hence, a new column has to be added to T_1. T_1 further goes through horizontal and vertical expansion. A copy of the current state table has been further appended to the end of the current T_1 (Step 7), which makes the size of T_1 to be $(8, 3)$ (Step 8). The protein d has now been added to T_1 in the form of a new column, and populated with 0's and 1's as given in Step 9. The horizontal expansion has made T_1 to be of size $(8, 4)$ (Step 10). S_1 has been updated to have the elements a, b, c, d. Since the interaction type is *binding/association*, all the states in T_1 satisfy the interactions rules, hence no elimination is needed. Therefore, the current size of T_1 is $(8, 4)$. Since no more interactions are left to be processed, the current state table is the final stable state table, where all the states r_1 through r_8 are stable.

After creation of the stable state table T_1 for the subnetwork G_1, the stable state table T_2 is formed for the subnetwork G_2 in the same way. From the stable state tables, the table M is formed as depicted in Figure 7.2 (d). The columns s' and s'' represent the row numbers in either of T_1 or T_2 whose corresponding stable states are 1-Hamming distance apart. Hence, since $r_{1,1}$ and $r_{1,2}$ in T_1 are 1-Hamming distance apart, they have been added to M. From M, the graph G'_1 is formed, such that the vertices $v'_{1,1}, v'_{1,2} \ldots v'_{1,8}$ of the vertex set V'_1 correspond to the stable states $r_{1,1}, r_{1,2} \ldots r_{1,8}$ of T_1. Likewise, the graph G'_2 is formed, such that the vertices $v'_{2,1}, v'_{2,2}, v'_{1,3}$ of the vertex set V'_2 represent the stable states $r_{2,1}, r_{2,2}, r_{2,3}$ of T_2. The edges are represented by the pair of states as given in Figure 7.2 (d). For example, the vertices $v'_{1,1}$ and $v'_{1,4}$ belonging to subnetwork G'_1 form an edge (row 1), and so does the pair of vertices $v'_{2,1}$ and $v'_{2,2}$ (row 11) belonging to subnetwork G'_2. Thus, we have got G'_1 and G'_2 (Figure 7.2 (e)).

In G'_1, three cycles of length 4 each, two cycles of length 6 each and one cycle of length 8 (Figure 7.2 (e)) have been formed. These cycles are $r_{1,2} - r_{1,3} - r_{1,7} - r_{1,6} - r_{1,2}, r_{1,1} - r_{1,4} - r_{1,8} - r_{1,5} - r_{1,1}, r_{1,6} - r_{1,5} - r_{1,1} - r_{1,2} - r_{1,6}, r_{1,1} - r_{1,2} - r_{1,3} - r_{1,7} - r_{1,6} - r_{1,5} - r_{1,1}, r_{1,2} - r_{1,1} - r_{1,4} - r_{1,8} - r_{1,5} - r_{1,6} - r_{1,2}$ and $r_{1,3} - r_{1,7} - r_{1,6} - r_{1,5} - r_{1,8} - r_{1,4} - r_{1,1} - r_{1,2} - r_{1,3}$. In G'_2, no cycles can be formed.

In order to demonstrate the change in $Rscore$ due to the change of a value of protein in the network/subnetwork, let us consider the subnetwork G_1 (Figure 7.2 (a)) in a stable state $r_{1,1}$ (0000). Consider the cycle $v'_{1,1} - v'_{1,5} - v'_{1,8} - v'_{1,4} - v'_{1,1}$. Suppose a unit of noise is introduced in the network such that the value of protein d is changed from 0 to 1. Now the new state becomes 0001 which is a stable state and within the cycle. Hence, the network is able to withstand the one unit noise/change and remain in the same cycle. Now suppose instead of d, the protein a is perturbed, i.e., the value of a is changed from 0 to 1. Hence the new state of the network becomes 1000. However, this state is an inconsistent state since it is not in the stable state table T_1. Therefore, the network is no longer stable. Since the stages of a biological process are represented by these stable states in the cycles, it is crucial that the network/subnetwork returns to one of the states within the cycle even after perturbation. Even if the network/subnetwork makes transition from one cycle to another due to the one unit noise/change, the network as a whole remains stable.

Before perturbation of the network G, the number of stable states for subnetwork G_1 is 8 (m_1) while the total number of possible states is 16 ($= 2^4$, where $n_1 = 4$). The $Rscore$ of the sample subnetwork G_1 given in Figure 7.2 (a) is 0.5. Considering the subnetwork G_2, it has 3 (m_2) stable states and the total number of possible states is 8 ($= 2^3$, where $n_2 = 3$). Hence, $Rscore$ of G_2 is 0.375. Therefore, the $Rscore$ of network G is 0.4375 (=(0.5+0.375)/2). Suppose a toxin, say t, is introduced into the sample network G, which inhibits proteins b and d in G_1 as given in Figure 7.4. As observed from the figure, after perturbation the number of stable states is 10 (m'') while the total number of proteins n''_1 $(= n_1 + 1)$ in the perturbed subnetwork G_1 is 5. Therefore, $PRscore$ for subnetwork G_1 is 0.3125 . Before perturbation, the sample subnetwork G_1 had an $Rscore$ of 0.5. Since no proteins are perturbed in subnetwork G_2, its $PRscore$ is the same as its $Rscore$, i.e., 0.375.


Perturbation of G_1 by toxin t
T_1
Redundant copying
Removing inconsistent states
New column for toxin t added to T_1

Row	State	a	b	c	d
1	$r_{1,1}$	0	0	0	0
2	$r_{1,2}$	0	1	0	0
3	$r_{1,3}$	1	1	0	0
4	$r_{1,4}$	0	0	1	0
5	$r_{1,5}$	0	0	0	1
6	$r_{1,6}$	0	1	0	1
7	$r_{1,7}$	1	1	0	1
8	$r_{1,8}$	0	0	1	1

T_1 before perturbation
Number of stable states=8
Number of nodes=4

Rscore = (0.5 +0.375)/2
= 0.4375

Row	State	a	b	c
1	$r_{2,1}$	0	0	0
2	$r_{2,2}$	0	0	1
3	$r_{2,3}$	1	1	1

T_2 before perturbation
Number of stable states=3
Number of nodes=3

G_1 (T_1) perturbed by toxin *t*. *t* inhibits *b* and *d*.

State	a	b	c	d
$r_{1,1}$	0	0	0	0
$r_{1,2}$	0	1	0	0
$r_{1,3}$	1	1	0	0
$r_{1,4}$	0	0	1	0
$r_{1,5}$	0	0	0	1
$r_{1,6}$	0	1	0	1
$r_{1,7}$	1	1	0	1
$r_{1,8}$	0	0	1	1

State	a	b	c	d
$r_{1,1}$	0	0	0	0
$r_{1,2}$	0	1	0	0
$r_{1,3}$	1	1	0	0
$r_{1,4}$	0	0	1	0
$r_{1,5}$	0	0	0	1
$r_{1,6}$	0	1	0	1
$r_{1,7}$	1	1	0	1
$r_{1,8}$	0	0	1	1
$r_{1,9}$	0	0	0	0
$r_{1,10}$	0	1	0	0
$r_{1,11}$	1	1	0	0
$r_{1,12}$	0	0	1	0
$r_{1,13}$	0	0	0	1
$r_{1,14}$	0	1	0	1
$r_{1,15}$	1	1	0	1
$r_{1,16}$	0	0	1	1

State	a	b	c	d	t
$r_{1,1}$	0	0	0	0	0
$r_{1,2}$	0	1	0	0	0
$r_{1,3}$	1	1	0	0	0
$r_{1,4}$	0	0	1	0	0
$r_{1,5}$	0	0	0	1	0
$r_{1,6}$	0	1	0	1	0
$r_{1,7}$	1	1	0	1	0
$r_{1,8}$	0	0	1	1	0
$r_{1,9}$	0	0	0	0	1
$r_{1,10}$	0	1	0	0	1
$r_{1,11}$	1	1	0	0	1
$r_{1,12}$	0	0	1	0	1
$r_{1,13}$	0	0	0	1	1
$r_{1,14}$	0	1	0	1	1
$r_{1,15}$	1	1	0	1	1
$r_{1,16}$	0	0	1	1	1

State	a	b	c	d	t
$r_{1,1}$	0	0	0	0	0
$r_{1,2}$	0	1	0	0	0
$r_{1,3}$	1	1	0	0	0
$r_{1,4}$	0	0	1	0	0
$r_{1,5}$	0	0	0	1	0
$r_{1,6}$	0	1	0	1	0
$r_{1,7}$	1	1	0	1	0
$r_{1,8}$	0	0	1	1	0
$r_{1,9}$	0	0	0	0	1
$r_{1,10}$	~~0~~	~~1~~	~~0~~	~~0~~	~~1~~
$r_{1,11}$	~~1~~	~~1~~	~~0~~	~~0~~	~~1~~
$r_{1,12}$	0	0	1	0	1
$r_{1,13}$	~~0~~	~~0~~	~~0~~	~~1~~	~~1~~
$r_{1,14}$	~~0~~	~~1~~	~~0~~	~~1~~	~~1~~
$r_{1,15}$	~~1~~	~~1~~	~~0~~	~~1~~	~~1~~
$r_{1,16}$	~~0~~	~~0~~	~~1~~	~~1~~	~~1~~

State	a	b	c	d	t
$r_{1,1}$	0	0	0	0	0
$r_{1,2}$	0	1	0	0	0
$r_{1,3}$	1	1	0	0	0
$r_{1,4}$	0	0	1	0	0
$r_{1,5}$	0	0	0	1	0
$r_{1,6}$	0	1	0	1	0
$r_{1,7}$	1	1	0	1	0
$r_{1,8}$	0	0	1	1	0
$r_{1,9}$	0	0	0	0	1
$r_{1,10}$	0	0	1	0	1

Stable state table after perturbation

PRscore = (0.3125+0.375)/2
= 0.3437

Figure 7.4: The diagram depicts the flow of BNRA when the hypothetical pathway in Figure 7.2 is perturbed by toxin t.

$Rscore$ of the network G (combining $Rscore$ of G_1 and G_2) is 0.4375. $PRscore$ of the network G (combining $PRscore$ of G_1 and G_2) is 0.3437. The difference in the $Rscore$ and $PRscore$ depicts the change in robustness of the network. Hence, a drop in the stability of the network G from 0.4375 to 0.3437 is observed due to the perturbation. The mathematical validation of the algorithm is given in Section 3. The analysis of time complexity of BNRA, that has been found to be $\mathscr{O}(\alpha 2^{4\beta_l})$, has been furnished in Section 4.

7.3 Mathematical validation

In this section, we describe the mathematical basis for some of the steps of the algorithm BNRA.

Lemma 1: The final stable state table obtained by the algorithm Boolean logic-based Network Robustness Analyzer (BNRA) is independent of the order of interactions processed by

BNRA.

Proof: Each of the interactions can be represented by Boolean expression obtained from a truth table. For example, the interaction x *activation* y can be represented by a truth table with the output $00 = \bar{x}\bar{y}$, $01 = \bar{x}y$, $11 = xy$. The expression obtained by deriving the sum of products (SOP) form is $\bar{x}\bar{y} + \bar{x}y + xy$, minimization of which leads to the expression $\bar{x} + y$. In a similar way, x *inhibition* y can be represented as $\bar{x} + \bar{y}$, while x *ubiquitination* y can be represented as $xy + \bar{x}\bar{y}$. Since *binding/association* consists of all possible combinations of 0's and 1's, therefore x *binding/association* y is represented as 1.

Intuitively, BNRA generates stable states for a network by incorporating all its interactions. This means, for a sample network with three interactions, say i, j, k, the stable state table has been generated by taking into consideration all the three interactions. The stable state table generated by BNRA is assumed to be a truth table. It can be represented by a boolean expression, which is obtained by ANDing the individual boolean expressions of each of the interactions. In boolean algebra, the associative law states that the AND operation (product) for a group of expressions can be done in any order. Considering expressions x', y', z', according to the law, their product is associative, i.e., $(x'y')z' = x'(y'z')$. Generating the stable state table of a network with β interactions indicates a chain of AND operations for β expressions. Hence, under associative law, the final stable state table is independent of the order of execution of the interactions.

Let us consider the sample subnetwork G_2 given in Figure 7.2. The interactions *e ubiquitination f* and *f activation g* are represented by expressions $ef + \bar{e}\bar{f}$ and $\bar{f} + g$. ANDing of the two expressions in any order leads to the final expression $\bar{e}\bar{f} + efg$. We form a table with all the 2^3 stable states and then eliminate the spurious and inconsistent states with respect to the interactions, considering all the interactions simultaneously as given in Figure 7.5. It is observed that the final stable state table T_2 is the same for Figures 7.3 (b) and 7.5. To be more certain, we derive the expression for the final stable state table in Figure 7.5 in sum of product form. The expression obtained is $\bar{e}\bar{f}\bar{g} + \bar{e}\bar{f}g + efg$. Minimization of the above expression leads to the final expression $\bar{e}\bar{f} + efg$, which is identical to the expression obtained by ANDing the expressions of interactions. Hence, we can conclude that the order of processing the interactions has no effect on the state of final stable state table.

Lemma 2: If an interaction is of the type *binding/association* and the two proteins involved in the interaction are already in S, then executing the redundant copying and the elimination steps for this interaction can be skipped.

Proof: The interaction *binding/association* has been represented by all possible combinations of 0/1. Suppose, at an intermediate stage, a new interaction has to be included in the state table T_l. Let us assume that the two proteins involved in this interaction are already in S

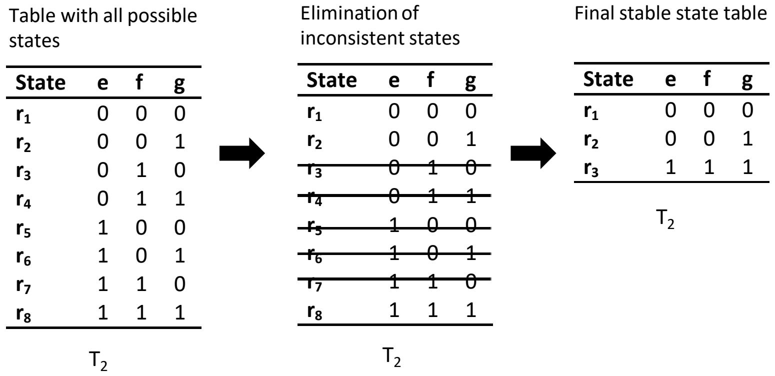


Table with all possible states

State	e	f	g
r_1	0	0	0
r_2	0	0	1
r_3	0	1	0
r_4	0	1	1
r_5	1	0	0
r_6	1	0	1
r_7	1	1	0
r_8	1	1	1

T_2

Elimination of inconsistent states

State	e	f	g
r_1	0	0	0
r_2	0	0	1
~~r_3~~	~~0~~	~~1~~	~~0~~
~~r_4~~	~~0~~	~~1~~	~~1~~
~~r_5~~	~~1~~	~~0~~	~~0~~
~~r_6~~	~~1~~	~~0~~	~~1~~
~~r_7~~	~~1~~	~~1~~	~~0~~
r_8	1	1	1

T_2

Final stable state table

State	e	f	g
r_1	0	0	0
r_2	0	0	1
r_3	1	1	1

T_2

Figure 7.5: The diagram depicts that the order of execution of interactions does not affect the state of the final stable state table.

and their values in T_l are represented by $c_{l,k}$ and $c_{l,k'}$. In such a case, the current combination of 0/1 for kth and k'th proteins are already valid since a binding interaction is represented by all possible combinations of 0/1, hence no redundant copying is necessary. Also, the entries in the state table for kth and k'th proteins are already consistent with all the other interactions already processed. Hence the elimination phase is redundant.

For example, consider a network with 3 interactions, ab (*activation*), bc (*inhibition*) and ac (*binding/association*). Let us assume that the interactions have been processed in the order as mentioned above. The final set of stable states would be 001 and 110. The stable states before processing the interaction bc and after processing bc would be the same. Even if the interaction ac is processed before any of the other two interactions, the final result would remain the same. If the interaction ac were not processed at all, then the set of stable states would have been unchanged. Hence, the binding interaction can be skipped to process the next interaction in the list if both the proteins involved in the interaction are already present in S.

Lemma 3: The execution time of BNRA depends on the order of execution of *binding/association* from a list of interactions.

Proof: The operation of redundant copying is time-consuming. Larger the state table more is the time to copy. As explained in Lemma 2, for binding interactions, if both the proteins involved in the interaction are already in S, the binding operation can be skipped to process the next operation. If the binding interaction were processed before other types of interactions, it would have led to the two proteins of the *binding/association* interaction to be present in S_l list, thereby unnecessary copying operation would have been carried out. According to BNRA, for *binding/association*, the number of copies of the initial state table that needs to be made is either one (if only one protein is present in S) or three (if both the proteins are absent from S). The *binding/association* interaction, unlike non-binding ones, have no constraints (allows all possible combinations of 0's and 1's). Presence of any other interaction involving the proteins of a binding interaction will overshadow the effect of *binding/association* interaction on the state table. Hence, the copying operation brought about by binding interaction would eventually be unnecessary if these two proteins are already a part of other interactions that have not yet been processed. Therefore, to reduce the processing time, in order to make sure that no extra copying operation is carried out, non-binding interactions are processed first.

Lemma 4: If every protein has a *binding/association* interaction with every other protein in a network (i.e., the graph is a completely connected graph with all interactions being *binding/association*), that network has a $Rscore$ of 1.
Proof: For every α subnetworks, if each of the n_l proteins has binding interactions with every other protein, the total number of stable states (m_l) would become 2^{n_l}. If this is the case for all the subnetworks, the total number of stable states become $2^{n_1}, 2^{n_2} \dots 2^{n_l}$. Consequently, $Rscore$ for each of the subnetworks will be $\frac{2^{n_1}}{2^{n_1}}, \frac{2^{n_2}}{2^{n_2}}, \dots \frac{2^{n_\alpha}}{2^{n_\alpha}}$. Hence the final $Rscore$ for the network will be unity.

Lemma 5: $Rscore$ is independent of the number of binding interactions.
Proof: Let us consider a connected network G with β interactions and n proteins. It has a stable state table T with m proteins. Hence $Rscore$ is $\frac{m}{2^n}$.

Case 1: Let a new *binding/association* interaction be added to the network, such that one of the proteins in the interaction is already in the network. For such a case, the number of stable states is $2 \times m$ (where m is the number of stable states before the new *binding/association* interaction is added). Total number of proteins before the introduction of the new interaction is n. Since one of the proteins of the new interaction is already in the network, the updated number of proteins is $n + 1$. Hence, the total number of allowable states after the new interaction is introduced is 2^{n+1}. Hence, the new $Rscore$ is $\frac{2 \times m}{2^{n+1}}$ which is equivalent to $\frac{2 \times m}{2^n \times 2}$; thereby the final $Rscore$ is $\frac{m}{2^n}$. Thus, $Rscore$ is independent of *binding/association* interactions in which one of the proteins associated with these interactions, is already a part of some other interaction in the network.

Case 2: Let a new *binding/association* interaction be added to the network, such that, both the proteins in the interaction is not already present in the network. In this case, the new number of stable states is $4 \times m$. Since two new proteins have been added to the network, the new number of allowable stable states is 2^{n+2}. Hence, the new $Rscore$ is $\frac{4 \times m}{2^{n+2}}$, which is equivalent to $\frac{4 \times m}{2^n \times 4}$. Therefore, the final $Rscore$ is $\frac{m}{2^n}$. Thus, $Rscore$ is also independent of *binding/association* interactions, where the proteins associated with them do not previously exist in the network.

Case 3: Let a new *binding/association* interaction be added to the network, such that both the proteins in the interaction already exists in the network. According to Lemma 2, no redundant copying or elimination operation is needed. Hence, the final $Rscore$ is $\frac{m}{2^n}$. $Rscore$ is, therefore, independent of *binding/association* interactions where both the proteins associated with these interactions, already exists in the network.

As described in the three cases above, $Rscore$ of a network is independent of the number of *binding/association* interactions. However, *binding/association* interactions are crucial in determining all the stable states of the network.

7.4 Analysis of Time Complexity

BNRA takes a network G as its input. The network has α subnetworks where each subnetwork contains β_l interactions and n_l proteins, where $1 \leq l \leq \alpha$. The first module of BNRA is fragmentation. In order to determine the disconnected subnetworks of an undirected network, we have used a depth-first search algorithm. The run-time for fragmentation of the initial network using depth first search algorithm is $|V| + |E|$ which is equivalent to $(n_1 + n_2 + \cdots + n_\alpha) + (\beta_1 + \beta_2 + \cdots + \beta_\alpha)$ where $|V| = n_1 + n_2 + \cdots + n_\alpha, |E| = \beta_1 + \beta_2 + \cdots + \beta_\alpha$. Thus the computational complexity for fragmentation is $\mathscr{O}(n + \beta)$, where $n = n_1 + n_2 + \cdots + n_\alpha$ and $\beta = \beta_1 + \beta_2 + \cdots + \beta_\alpha$.

The next module of BNRA is initialization. In this operation, the initial state table for each of the α subnetworks is created. The state table has either 3 rows (for non-binding interactions) or 4 rows (for binding interactions). Hence this step has a time complexity of $\mathscr{O}(4\alpha)$ which is equivalent to $\mathscr{O}(\alpha)$.

Initialization is followed by redundant copying. In this step, multiple copies of the current state table are formed and appended to the end of the current state table. The maximum number of states for any type of interaction is 4 (*binding/association*). According to Lemma 2, the size of the stable state table will be maximum, when all the interactions are of the type *binding/association*. Let c_p be the number of copy operations (number of individual elements of the table to be replicated) required for pth interaction, such that $1 \leq p \leq \beta_l$. We will derive the complexity of this module in 4 cases.

Case 1: If the network has one interaction ($\beta_l = 1$), maximum number of proteins will be 2, hence $n_l = 2$. The maximum number of stable states will be 2^{n_l} hence $m_l = 4$, therefore $c_1 = 0$. In this case, there is no copy operation required.

Case 2: For $\beta_l = 2$ interactions, the maximum number of proteins will be $n_l = 4$ ($n_l = 2\beta_l$, two times the number of interactions, provided none of the proteins associated with the interactions are common). The maximum number of stable states will be 2^{n_l}. The number of copy operations would be equal to the number of elements being copied. The number of elements to be copied will be equal to the number of rows times the number of columns of the current state table. For first interaction, $n_l = 2, m_l = 4$, no copy operation is performed, therefore $c_1 = 0$. For second interaction, with the current size of table being $n_l = 2$ and $m_l = 2^2$, the number of copies to be made is 3, thus $c_2 = m_l \times n_l \times 3$. The total number of copy operations is $c_1 + c_2$ which is equivalent to $0 + m_l \times n_l \times 3$. Replacing the values of m_l and n_l we get the total number of copy operation, which is $(2^2 \times 2 \times 3)$.

Case 3: For $\beta_l = 3$ interactions, the maximum number of proteins will be $n_l = 6$ ($n_l = 2\beta_l$). The maximum number of stable states will be 2^{n_l}. For first interaction, $n_l = 2, m_l = 4$, no copy operation is performed, therefore $c_1 = 0$. For second interaction, with the current maximum size of table being $n_l = 2$ and $m_l = 2^2$, the number of copies to be made is 3, hence $c_2 = m_l \times n_l \times 3$. Replacing m_l, n_l with their values, we get $c_2 = (2^2 \times 2 \times 3)$. For third interaction, with the current maximum size of table being $n_l = 4$ and $m_l = 2^4$, the number of copies to be made is 3. Hence, $c_3 = m_l \times n_l \times 3$ which is equivalent to $c_3 = (2^4 \times 4 \times 3)$. Hence, the total number of copy operations is $c_1 + c_2 + c_3$ which is equivalent to $0 + (2^2 \times 2 \times 3) + (2^4 \times 4 \times 3)$. In the same way, for $\beta_l = 4$, the total number of copy operations is $c_1 + c_2 + c_3 + c_4$ which is equivalent to $0 + (2^2 \times 2 \times 3) + (2^4 \times 4 \times 3) + (2^6 \times 6 \times 3)$, where $n_l = 6$ and $m_l = 2^6$.

Case 4: For β_l interactions, with $n_l = 2\beta_l, m_l = 2^{2\beta_L}$, the total number of copy operations is $0 + (2^2 \times 2 \times 3) + (2^4 \times 4 \times 3) + (2^6 \times 6 \times 3) \ldots (2^{2(\beta_l - 1)} \times 2(\beta_l - 1) \times 3)$. This is equivalent to $3 \sum_{i=1}^{\beta_l} (2^{2(i-1)} \times 2(i-1))$. A simplified version of the expression would be $3 \sum_{i=1}^{\beta_l} (2^{2i-1} \times (i-1))$. Therefore, the time for copy operation for β_l interactions cannot be more than two times the time required for copy operation for β_lth interaction, which is equivalent to $3(2^{2\beta_l})(\beta_l - 1)$. Hence this step has a time complexity of $\mathscr{O}(2^{2\beta_l}(\beta_l - 1))$.

In elimination step, checking whether a state is to be eliminated or not takes n unit of time, where n is the number of proteins in the stable state table. All the states in the current stable state table need to be checked. The maximum number of states at any point in time is $2^{2\beta_l}$. The elimination operation is carried out $\beta_l - 1$ times, i.e., every time a new interaction is taken into consideration. Therefore, for lth module, the maximum time for checking is $2^2 + 2^4 + \ldots + 2^{2\beta_l}$ which is smaller than $2^{2\beta_l + 1}$. Hence, for all the α subnetworks, this module can take the maximum time of $\mathscr{O}(\alpha 2^{2\beta_l + 1})$.

The next step is robustness calculation. In this step, we have created a table M for α subnetworks, which contains a list of stable state pairs that are 1-Hamming distance apart. Since each stable state has been compared with all the other states to determine if they are 1-Hamming distance apart, the complexity to create this table is $\mathscr{O}(\alpha(2^{2\beta_l})^2)$ which is equivalent to $\mathscr{O}(\alpha 2^{4\beta_l})$, provided the total number of possible stable states is $2^{2\beta_l}$. The score calculation module, which performs calculation of score in constant time, is of computational complexity $\mathscr{O}(\alpha)$ for all the subnetworks.

Hence, the total complexity of BNRA is $\mathscr{O}(n+\beta+\alpha+2^{2\beta_l}(\beta_l-1)+\alpha 2^{2\beta_l+1}+\alpha 2^{4\beta_l}+\alpha)$ which is equivalent to $\mathscr{O}(\alpha 2^{4\beta_l})$.

7.5 Results

We have applied BNRA on 221 pathways belonging to 26 different categories [209], which include signal transduction, transportation and metabolism, cell growth and death, and human diseases. Figure 7.6 displays the distribution of $Rscore$. As observed from these figures, $Rscore$ for most of the pathways lie between 0.3 and 0.6. Four pathways have $Rscore$ greater than 0.9, indicating that they are the most stable ones over the others, signifying high robustness to perturbations. In this section, we give an insight into the $Rscore$ and $PRscore$ derived from 221 pathways using BNRA. First, we give a general overview of the application of BNRA on these pathways. This is followed by a detailed analysis of the same on disease pathways. Following this is the study of effect of perturbation on signaling networks, which has been supported by biological evidence. Finally, a comparison of BNRA with other existing algorithms has been provided. The detailed results of application of BNRA on these 221 pathways belonging to 26 groups have been provided in Table A.16 in Appendix A. A summary of the average $Rscore$, $PRscore$ and the percentage fall of $Rscore$ for each of the 26 groups have been given in Table A.17 of Appendix A.

7.5.1 Application of BNRA on 221 pathways

The 221 pathways obtained from KEGG can be grouped into 26 groups depending upon their functionality. The average $Rscore$, $PRscore$ and the drop in the stability of each of these groups of pathways have been determined. The genetic information processing group consists of 6 pathways, having an average $Rscore$ of 0.5807, and a $PRscore$ of 0.3026, with an average of 47% fall in stability on perturbation. The signal transduction group comprises 30 pathways, the largest number of pathways in any group. It has resulted in an $Rscore$ of 0.3858, indicating a group of low stability pathways. This group of pathways has reported a $PRscore$ of 0.1660, with an average of 56% fall in the stability after being perturbed. The maximum fall in the stability of 77.79%, however, has been recorded for the group of pathways under development and regeneration of axon cells. However, this set of pathways has reported low stability having an $Rscore$ of 0.3637. The lowest drop of 23% in stability has been reported by pathways under the group of antimicrobial drug resistance. The neurodegenerative disease pathways, consisting of four pathways, have reported an average $Rscore$

of 0.5349 and a $PRscore$ of 0.2787, resulting in the drop of stability of the network by 47%.

Among the disease pathways, cancer pathways have reported the maximum average stability with an $Rscore$ of 0.6333. Cancer pathways have resulted in a drop in stability by an average of 44%. Apart from these, the cellular pathways for eukaryotes and prokaryotes have resulted in a similar $Rscore$ of 0.3275 and 0.3392 respectively. Pathways conforming transport and catabolism in cells have reported an average $Rscore$ of 0.6034 with an average $PRscore$ of 0.3364, indicating a stability drop by 44%. The lowest stability having an $Rscore$ of 0.3234 has been reported by endocrine and metabolic diseases. This group of pathways has reported a drop in stability by 69%, with an average $PRscore$ of 0.0987. Cell motility group of pathways have reported an $Rscore$ of 0.5198 and a $PRscore$ of 0.2147, denoting an average drop of 58% in the stability of the pathways on perturbation.

Apart from the aforesaid pathways, the endocrine system, circulatory system, digestive system, excretory system, nervous system and sensory system have resulted in $Rscore$'s of 0.4134, 0.3249, 0.4266, 0.5415, 0.3524, and 0.4436 respectively. The $PRscore$'s of these pathways, on perturbation, have been found to be 0.1918, 0.1597, 0.2076, 0.2801, 0.1957, and 0.19766, resulting in a drop of 53%, 50%, 51%, 48%, 44%, and 55%, respectively, in their stability. Pathways involving aging have also reported a huge drop in stability by 62%, having an $Rscore$ of 0.3550 and a $PRscore$ of 0.1331.

The variation of $Rscore$ over different categories of these 221 pathways has been depicted in Figure 7.7. As observed from the figure, pathways involved in nervous system have shown maximum variation in $Rscore$. This indicates that the stability of the nervous system pathways varies in the interval $[0.0016, 1]$. On the other hand, developmental pathways have shown minimum variation in $Rscore$ (0.2838 $\leq Rscore \leq$ 0.4312).

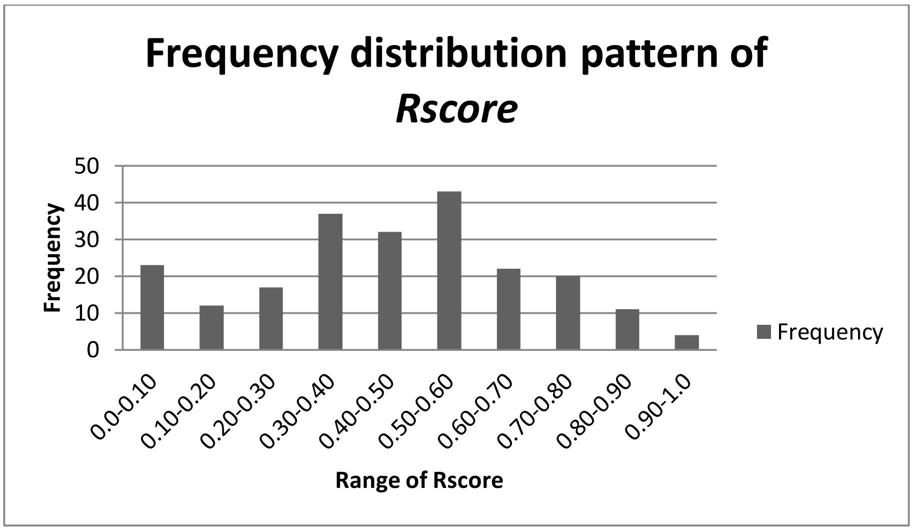


Figure 7.6: Histogram of $Rscore$ for 221 pathways.

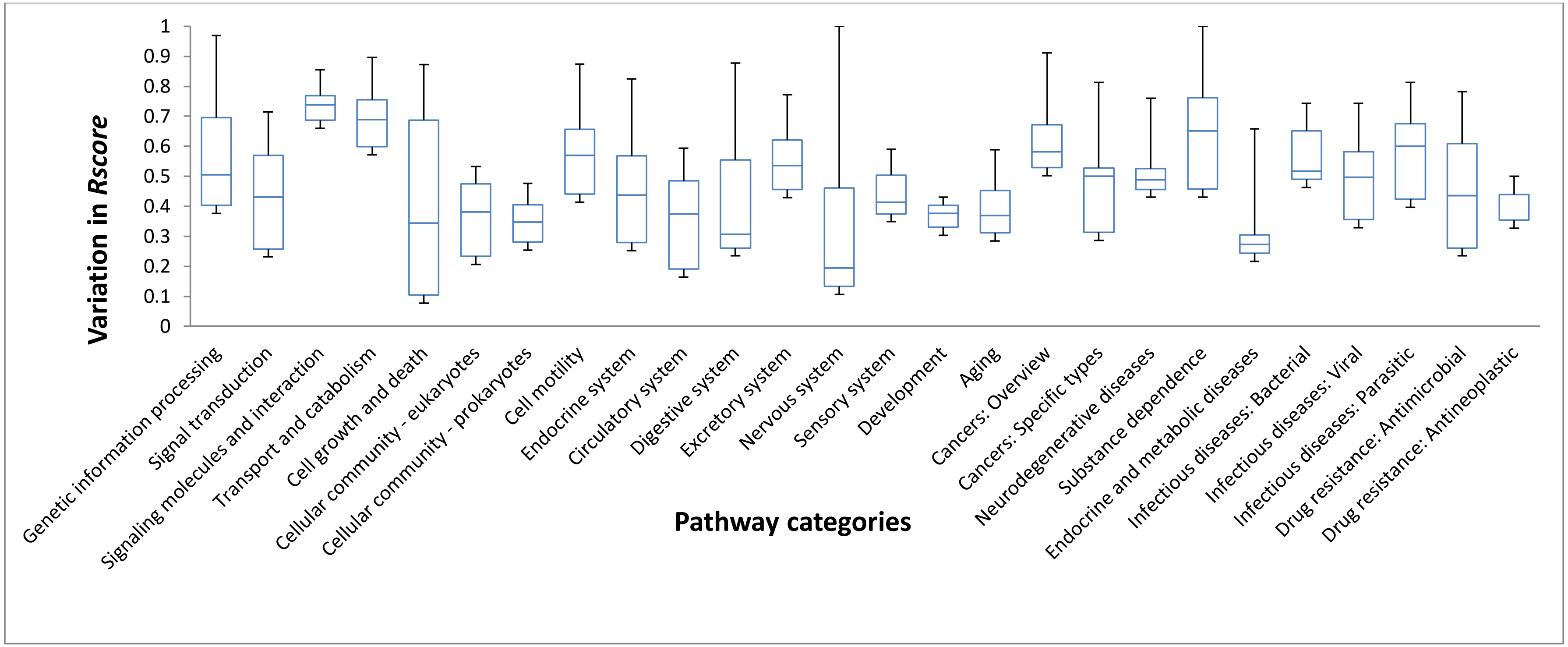


Figure 7.7: Variation of $Rscore$ over different categories of 221 pathways.

7.5.2 Effect of perturbation on signaling networks and their biological validation

Here, we study the effect of perturbation on various signal transduction pathways, followed by validation of the results based on existing literature. For each of these 221 pathways, we have recorded $PRscore$ on perturbing each of the proteins. For each protein in a pathway, $PRscore$ obtained for perturbing it has been recorded. We have noted the lowest $PRscore$ among all the scores generated. The average difference between $Rscore$ and the lowest $PRscore$ for each of the 26 groups of pathways has been depicted in Figure 7.8.

It has been observed that the maximum perturbation is caused in the Cytosolic DNA-sensing pathway ($Rscore$ reduced by 74.04%) while the minimum effect has been observed in Neuroactive ligand-receptor interaction ($Rscore$ having been reduced by 0.94%). Figure 7.8 depicts the summarized effect of perturbation on all the 26 categories of 221 pathways. Higher the difference between $Rscore$ and $PRscore$ due to perturbation, more prone it is to disruption of the pathway. As observed from the figure, the development and regeneration pathways are the least robust among the groups of pathways. The drug resistance pathways for antimicrobial infections have been found to be the most stable against perturbations.

- **Ferroptosis pathway** - The ferroptosis pathway has generated an $Rscore$ of 0.8164. In the pathway, perturbation of protein SLC7A11 (protein solute carrier family 7 members 11) has resulted in an $PRscore$ of 0.714 [430]; thus dropping the stability of the network. Perturbation of protein GPX4 (glutathione peroxidase 4) has led to a $PRscore$ of 0.684, resulting in a more significant drop. Biologically, perturbation of GPX4 is known to trigger acute renal failure in mice [15]. When protein TFRC (transferrin receptor) is perturbed, the stability of the network has further fallen to 0.652 [63].

- **Intestinal immune pathway** - Perturbation in the intestinal immune pathway has not yielded a drastic change in the stability. The pathway has an $Rscore$ of 0.8164. The protein CD40 triggers the production of antibody IgA. Its perturbation has led to a $PRscore$ of 0.7747, recording a fall in the stability of the network [335]. Perturbation of protein CCR9 (C-C motif chemokine receptor 9) has resulted in $PRscore$ of 0.7539. The perturbation of protein TNFSF13 (tumor necrosis factor ligand superfamily member 13) has resulted in a $PRscore$ of 0.7904. Not much change in the stability of the network is noticed due to perturbation, indicating that the IgA production may not be affected to a great extent, as suggested by Yang *et al.* [435].

- **Thyroid hormone synthesis pathway** - The thyroid hormone synthesis pathway is responsible for regulating thyroid level in the body. The pathway has resulted in an $Rscore$ of 0.824. Perturbation of the proteins PRKACA (protein kinase cAMP-

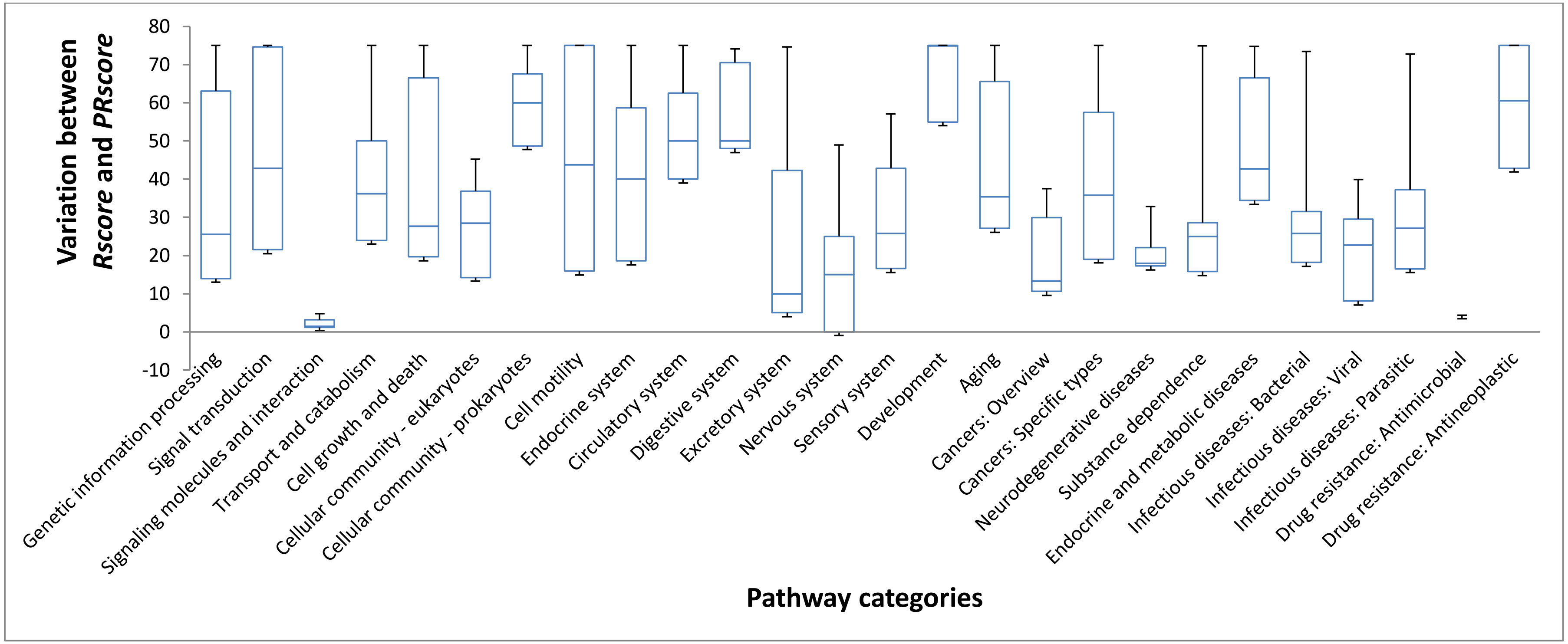


Figure 7.8: Variation in the difference of $Rscore$ and $PRscore$ in 221 pathways

activated catalytic subunit alpha), DUOX2 (dual oxidase 2), and TG (thyroglobulin) has resulted in $PRscore$ of 0.706, 0.6524 and 0.679 respectively. Ohara *et al.* have shown that perturbation of PRKACA protein has a detrimental effect on the pathway [300], which is also depicted by $PRscore$. Moeno *et al.* have concluded that perturbation of DUOX2 protein leads to hypothyroidism [284]. The drop in $Rscore$ due to its perturbation suggests that the stability of the network gets affected, which has led to such a consequence.

- **GABAergic synapse pathway** - The GABAergic synapse pathway plays a crucial role in normal function and long-term homeostasis of the neuronal circuit. It has resulted in an $Rscore$ of 0.7518. The protein GABRA1 (gamma-aminobutyric acid type A receptor alpha1 subunit) directly controls the efficiency of the GABAergic synaptic pathway, hence is largely responsible for maintaining the stability of the pathway [263]. When the protein has been perturbed, $Rscore$ has been reduced by 25% to $PRscore$ of 0.563. This drastic change in its stability is consistent with the claim made by Luscher *et al.* [263].

- **Endocytosis pathway** - EHD1 protein in the endocytosis pathway is required for the recycling of MHC class I molecules [159]. In the endocytosis pathway, the protein EHD1 (EH domain containing 1) has been inhibited, which has resulted in a drop of $Rscore$ (0.8958) to $PRscore$ of 0.770; hence indicating a reduction in the stability of the network. This is at par with the investigation of Yap *et al.* [441].

7.5.3 Application of BNRA on disease pathways and their biological validation

BNRA has analyzed 73 disease pathways out of the above 221 pathways, which mediate the development of cancers, viral infections, bacterial infections, among others. We have been able to biologically validate the results of execution on some of these 73 pathways, for which information is available in the literature.

- **Parkinson's disease** - Parkinson's disease is a long-term degenerative disorder of the central nervous system that mainly affects the motor system. On medical terms, no cure has been found for Parkinson's disease as of yet [245]. However, medications have been used to control them. Parkinson's disease has been reported to have $Rscore$ of 0.7604, indicating it to be a stable pathway. Casp9 is reported to be a core protein in Parkinson's disease [449]. The perturbation of Casp9 has resulted in a $PRscore$ of 0.7161, a visible drop from $Rscore$ to $PRscore$. UBA1 (ubiquitin-like modifier activating enzyme 1) is a crucial enzyme in Parkinson's disease. Parkinson's disease pathway has shown a 12% decrease in the stability of the network from $Rscore = 0.7604$

to $Rscore = 0.592$, when protein UBA1 is perturbed. UBA1 expression or activity may decrease with age. This may also lead to degeneration of specific sub-populations of neurons and/or affect the protein aggregation observed in these disorders [161].

- **COVID-19** - Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), also known by the provisional name 2019-nCoV, is a single-stranded RNA virus. It is contagious in humans and causes severe respiratory disorder, and in severe cases, death. BNRA has analyzed the pathway of infection due to the recently discovered novel coronavirus and the pathway of the immune response to the infection caused by COVID-19. Application of BNRA to the disease pathway [369] has resulted in an $Rscore$ of 0.5, indicating the pathway is moderately stable. Application of BNRA to the pathway of human immune system during infection caused by SARS-CoV-2 [252] has resulted in an $Rscore$ of 0.0074. Thus, it is not a very stable pathway. Perturbation of protein orf1ab in the disease pathway has led to $PRscore$ of 0, indicating a total disruption of the disease pathway. Therefore, protein orf1ab can be a possible drug target. In the pathway of immune response to the disease, it has been noticed that if the protein CLR is turned on and simultaneously any other protein in the pathway is turned off, the network's stability becomes 0, i.e., $PRscore$ is 0. In such a case, the immune system is said to have failed to control the infection caused by the virus.

- **Transcriptional misregulation in cancer** - Transcriptional misregulation refers to regulation that has gone awry from the normal or healthy state. It alters the expression (switching 'on' or 'off') of the genes in healthy cells. With a score of 0.9107, it is a stable pathway, indicating that the changes in expression that turns normal cells into cancer cells are stable and long term. Interestingly, the activation of transcriptional misregulation in cancer cells has led to dropping of $Rscore$ to 0.696. The protein SPI1 (Spi-1 proto-oncogene) is known to regulate replication in cancer cells. On perturbation, it has led to a decrease in stability of the pathway with a $PRscore$ of 0.7265. Its perturbation has a detrimental effect on the spreading of cancer cells [167].

- **Rheumatoid arthritis** - This pathway has resulted in an $Rscore$ of 0.8125, indicating it to be a stable pathway. As confirmed by experiments [327], $PRscore$, on perturbation of protein CD80 (type I membrane protein), has been found to be 0.5; thus displaying a drop in the stability of the pathway to a great extent.

- **Malaria** - For malaria, $Rscore$ has been found to be 0.8125, which indicates that the disease pathway in the host due to malaria is stable. The pathway has seven four-sized cycles, 16 six-sized cycles, and six eight-sized cycles. However, when protein MYD88 (innate immune signal transduction adaptor) is perturbed, the stability of the network has fallen to 0.5 with the number of four-sized cycles turning out to be two, while other cycles of sizes six and eight have ceased to exist.

- **Pathways in cancer** - Pathways in cancer have resulted in an $Rscore$ of 0.6721. Protein HRAS (transforming protein p21) in cancer pathways plays an important role in inducing cancer. Perturbing the protein has led to a drastic drop in the stability ($PRscore = 0.3124$) of the pathway as validated by Parikh *et al.* [310].

- **Choline metabolism in cancer** - Abnormal choline metabolism is emerging as a metabolic hallmark, which is associated with oncogenesis and tumor progression. The pathway has been reported to have an $Rscore$ of 0.5. According to Klein *et al.*, the protein PC-PLD (phosphatidylcholine-specific phospholipase D) is a crucial component of the pathway [219]. Perturbation of the same has led to a $PRscore$ of 0.1214. Thus, a visible drop in stability is noticed, indicating that the perturbation may weaken the pathway.

- **Alzheimer disease** - Alzheimer disease (AD) pathway has been reported to have an $Rscore$ of 0.4444. According to Helisalmi *at al.* [185], protein PSEN1 (Presenilin 1) is crucial for mediating the disease. BNRA has determined a $PRscore$ of 0.2315 when protein PSEN1 is perturbed; indicating a sharp drop in stability.

- **Tuberculosis -** Tuberculosis or TB pathway has reported an $Rscore$ of 0.4872. Scanga *et al.* have shown that protein MYD88 is majorly responsible for the resistance of the disease [345]. BNRA has performed perturbation of MYD88 protein, and $PRscore$ has been found to be 0.1973. As observed, the immune response pathway is significantly affected due to this perturbation.

7.6 Discussion on the comparative performance of BNRA with some existing algorithms

In this section, we have reported the comparative performance of BNRA with other investigations on a few pathways. All the models only approximate reality utilizing some formal representation. It is the degree to which BNRA approximates reality and to acquire knowledge about some physical phenomenon that forms the basis of the comparison. It has been noticed that the existing algorithms, developed by Davidich *et al.* [97], Gupta *et al.* [168], Flobak *et al.* [145], Fumia *et al.* [146] and Rodriguez *et al.* [333], have been able to generate the stable state table of only one particular pathway, and are not versatile in dealing with the different types of pathways. None of the existing investigations have developed a usable system for the readers, which would generate stable state tables and consequently, help them to perceive the effect of perturbation. The algorithm has been described, which the users will have to implement before it can be applied to pathways. BNRA has outperformed the existing algorithms in the following ways.

- **Cell cycle - yeast (sce04111):** Davidich *et al.* [97] have taken into consideration only the *activation* and *inhibition* interactions. The yeast cell cycle has 88 proteins in total, as given in KEGG. On the other hand, Davidich *et al.* have analyzed an incomplete network considering only 20 proteins. Apart from that, it has been reported that the pathway has 1024 (2^{10}) stable states, while BNRA has reported that the network has 128 stable states. This difference may be due to the consideration of only *activation* and *inhibition* interactions by Davidich *et al.*, while ignoring the other types of interactions. Incorporating all types of interactions increases the intricacy of the network and pushes the model closer to the actual biological scenario. With a diverse set of interactions, the number of stable states will supposedly be less. Hence, the analysis by BNRA is more realistic than that by Davidich *et al.* They have not derived any quantification measure for robustness.

- **Calcium signaling pathway (hsa04020):** Gupta *et al.* [168] have carried out a Boolean network analysis of neurotransmitter signaling pathway. In this investigation, they have considered the interactions of types *activation* and *inhibition*, whereas, the said biological pathway has interactions of type *phosphorylation* and *binding/association*. Hence, instead of 22 interactions, they have considered a simplified network with 18 interactions. On the other hand, BNRA has considered all the 22 interactions in the network. BNRA has found that the protein CALML6 (calmodulin like protein 6) activates and binds to seven other proteins, making it a hub protein. Gupta *et al.* has not reported such an observation.

- **Gastric cancer (hsa05226):** Flobak *et al.* [145] have considered gastric cancer as their logical model for predicting drug synergies in the pathway. It has taken 75 proteins into account, while our analysis has considered 80 proteins. The proteins PIK3CA (phosphatidylinositol-4,5-bisphosphate 3-kinase catalytic subunit alpha), CDK2 (cyclin dependent kinase 2), p27 (Cyclin-dependent kinase inhibitor), cyclin B1 (regulatory subunit of cyclin-dependent kinase 1) and cyclin A2 (regulator of the cell division) have not been considered by Flobak *et al.* However, these proteins, along with the proteins p53 (tumor protein), AKT1 (RAC-alpha serine/threonine-protein kinase), BCL2 (b-cell lymphoma 2) and MAPK1 (mitogen-activated protein kinase 1), have been identified as key targets for the treatment of gastric cancer [259]. The network considered by Flobak *et al.* is incomplete, making the results unreliable. On the contrary, BNRA has taken all the proteins into account, which have been identified as crucial for treatment of gastric cancer. Hence, the calculation of stability ($Rscore$) by BNRA is more reliable.

- **Pathways in Cancer (hsa05200):** Fumia *et al.* [146] have analyzed a simplified version of cancer pathway with 96 proteins and 249 interactions, while the original pathway considered by BNRA has 159 proteins and 169 interactions. The number of stable states obtained in Fumia *et al.* [146] is 32 million, while the number of states obtained by BNRA has been found to be 134 million. This difference may be due to two factors, one of them being the less number of proteins considered in the network by Fumia *et al.* compared to BNRA. Unlike BNRA, Fumia *et al.* have considered only *activation* and *inhibition*, another factor contributing to the low count of stable states.
- **Breast Cancer (hsa05224) pathways:** Rodríguez *et al.* [333] have analyzed the breast cancer pathways. They have taken 28 proteins pertaining to the pathway into account, but have been unable to generate the list of stable states since it is computationally infeasible. Besides, Rodríguez *et al.* have considered only *inhibition* and *activation* types of interactions, unlike BNRA. However, BNRA has analyzed the pathway considering 127 proteins and 120 interactions, taking all types of interactions into account. BNRA has successfully generated the list of stable states, which amounts to 134 million for the pathway.

7.7 Conclusions

In this chapter, we have developed an efficient algorithm, called Boolean logic-based Network Robustness Analyzer (BNRA). The algorithm quantifies the robustness of a network with a measure termed as $Rscore$. Robustness of biological pathways measured by $Rscore$ depicts its stability: a low score implying an unstable network vulnerable to be disrupted, a high score implying a pathway robust to perturbations. We have applied BNRA on 221 pathways, including 73 disease pathways. Whenever the robustness of the analyzed networks has been available in the literature, we used them to validate the scores obtained, and we found them to be in agreement. For example, among the disease pathways we analyzed, the transcriptional misregulation in cancer has been found to have the highest $Rscore$, indicating a highly stable network.

We then extended BNRA to also handle perturbation of networks. When a protein is perturbed, BNRA re-evaluates the entire network to determine the stability of the network on perturbation, resulting in a score that we call the $PRscore$. This score can then be used for insight into the effect of the perturbation on the stability of the resulting network. We also analyzed the pathways of the recently discovered SARS-CoV-2, including possible perturbations. We compared BNRA's performance with current state-of-the-art algorithms, which showed that BNRA generates a more comprehensive score, incorporating many more signals and their effects.

In the future, we would like to make it possible to analyze the effect of multiple simultaneous perturbations on various biological pathways and visualize as well as to measure them. We would also like to enhance the representation of biological networks by assigning direction to interactions of the network so that cause-effect relationships can be analyzed better.

Chapter 8

Conclusions and Scope for Future Research

This chapter summarizes the major contributions of each of the contributory chapters of the thesis. Additionally, it provides an insight into the scope for further work related to *in silico* identification of bacterial toxins and analyzing their effect on host pathways.

8.1 Major Contributions

Based on feature extraction, classification and pathway prediction, we have developed, in this thesis, novel algorithms and systems to facilitate the computational identification of bacterial toxins and analyzing their effect on host pathways. Predictions made by these algorithms and systems have been validated by corresponding experimental results available in literature. The prediction of effector proteins has been supported by existing *in vitro*/*in vivo* experiments that have identified various effectors in literature. Similarly, the algorithms on pathway prediction have been validated through appropriate *in vitro*/*in vivo* experimental results available in literature.

The thesis has started with a brief introduction of the basic concepts to enhance its readability, which constitutes Chapter 1. Chapter 2 is dedicated to the literature survey of host-pathogen interactions, encompassing analysis of the state-of-the-art procedures contributing towards the identification of toxins and their effect on host pathways. Here, a brief history of host-pathogen interactions and its classification has been provided, which are based on different factors, such as genes, proteins, host-factors, and inhibition mechanism of macrophages. We have analyzed various previously reported prediction methodologies on host-pathogen interactions. A spectrum of data repositories facilitating research in this domain has been elaborately discussed. From the survey, it has been safely concluded that the prediction of effector proteins is of prime importance, since these proteins are responsible for many diseases.

Chapter 3 deals with efficient prediction of T6 effector proteins based on their primary and secondary structures. Prediction of effector proteins from bacterial genome/proteome information is important for analysis of the role of their secretion systems in pathogenesis. Here we have developed a system, called PyPredT6, for *in silico* identification of T6 effector proteins. PyPredT6 extracts a set of 873 unique features from nucleotide and amino acid sequences of experimentally verified T6 effector proteins. Based on these features, ensemble learning has been carried out to predict whether an unknown protein is a T6 effector or not. It has successfully predicted 42 proteins out of 3850 proteins in *Yersinia pestis*, and 30 proteins out of 2736 proteins in *Vibrio cholerae* as T6 effectors. These predictions have further been validated by various experimental results reported in literature, which proves the effectiveness and reliability of PyPredT6. However, this investigation is restricted to primary and secondary structures. Additionally, the prediction of effectors has been limited to the whole proteome of two organisms.

Prediction of effector proteins would remain incomplete if their tertiary (3-dimensional) structure is not taken into consideration. Thus, in Chapter 4, we have developed an effector protein predictor system based on 3D structure (EPP3D) to identify various types of effector proteins using their 3D structural characteristics. We have taken into account a limited number of features (eight features) for effector prediction. In addition, a novel oversampling algorithm, called Cluster Quality-based Non-Reductional (CQNR) oversampling technique, has been developed in Chapter 4 to facilitate oversampling of effector protein datasets. Since the training dataset has been imbalanced, we have used CQNR to balance the effector protein dataset. On application of CQNR, considerable improvement has been reported in classification of samples in some benchmark datasets as well as various effector proteins. In order to demonstrate the effectiveness of EPP3D in effector protein classification, we also have considered a dataset derived from 3D structures of experimentally verified T3, T4, and T6 effector proteins. It has been noticed that tertiary structure-based classification results in an improved identification of effectors than classification based on primary and secondary structure only.

In Chapter 5, we have developed a deep neural network-based system, called DeepT7, to identify T7 effector proteins based on their primary and secondary structures. It may be mentioned here that we were unable to find tertiary structure of T7 effectors. The nucleotide and peptide sequences of experimentally verified effector proteins have been taken into consideration for constructing a set of 1727 unique features. This feature set has captured various aspects of effector proteins, which include their physicochemical properties, primary and secondary structure-based properties, and evolutionary information. Since the dataset is unbalanced, CQNR has been applied to it. A combination of these features and the aforesaid deep neural network-based framework has been used to perform *in silico* prediction of T7 effector proteins in *Mycobacterium bovis* and *Streptococcus pneumoniae* to ascertain the applicability of DeepT7. Experimental results, reported in literature, have proved the effectiveness and biological reliability of DeepT7.

On accomplishing the goal of successful development of various toxin prediction systems, next aim is the prediction and analysis of the effect of toxins on host pathways. In this regard, the novel algorithm developed in Chapter 6 has predicted unknown metabolic pathways from a pool of metabolites and the effect of toxin on such pathways. In other words, in Chapter 6, we have developed a novel algorithm, called Architectural Similarity-based Automated Pathway Prediction (ASAPP), which predicts biochemical transformations from 2D structure of metabolites. The algorithm enables us to predict the chance of a transformation of one metabolite to another, depending upon the 2D structural similarity of the metabolites and the difference in their molecular weights. Depending on these factors, a score has been assigned to each transformation. In addition, different threshold techniques have been applied to determine the final list of probable transformations. The *in silico* analysis has shown how the presence of a toxin in the host body may adversely affect its metabolic pathways. Here, we have predicted the effect of 52 such toxins on Glycolysis pathway and TCA cycle. The investigation has been conducted to explore the effect of individual toxins on these two pathways only. However, in the biological scenario, there are more than one pathway exposed to a toxin, and more than one toxin may be present in the host. For such a scenario, the prediction would have been different, and form a scope of future work.

Since metabolic and signal transduction pathways are both crucial for maintaining homeostasis in a host, the investigation would remain incomplete without exploring the effect of toxins on signal transduction pathways. In Chapter 7, we have developed an algorithm, entitled Boolean logic-based Network Robustness Analyzer (BNRA), for quantifying the robustness and analyzing the effect of toxins on signal transduction pathways. BNRA models biological pathways in the form of undirected graphs. The interactions among the proteins have not been assigned a direction since the interaction *binding/association* cannot be given a direction. Some other types of interactions, such as *missing*, *indirect effect* and *compound*, have not been incorporated in the investigation, since the effect of these interactions on proteins are unknown. BNRA has computed the robustness of both unperturbed and perturbed networks. It defines quantitative measures $Rscore$ and $PRscore$ to quantify the robustness of a network for both before ($Rscore$) and after ($PRscore$) perturbation. BNRA has been applied to 221 pathways belonging to 26 categories, including human disease networks, to analyze their characteristics. Among these 221 pathways, four of them, *viz.*, mRNA surveillance pathway, transcriptional misregulation in cancer, hypertrophic cardiomyopathy (HCM) and synaptic vesicle cycle, have an $Rscore$ greater than 0.9, indicating that these networks are the most robust ones among all the pathways under consideration. The drop in stability of pathways, which is defined by the difference in $Rscore$ and $PRscore$, has been derived, in order to have an insight into the extent to which the stability of pathways gets affected by perturbations. Some of the pathways like the COVID-19 pathway, MAPK signaling pathway, autophagy pathway in animals, and biofilm formation pathway in *Vibrio cholerae* have demonstrated a drastic drop in their stability, revealing their vulnerability.

In order to facilitate and encourage further research in this field, we have made the algorithms available to researchers in the form of executable applications, along with instructions guiding their usage. This will help in generating results with ease without implementing the algorithms.

8.2 Future Scope

In future, these algorithms can be improved in scope of their applicability and efficiency. Further, we would like to collaborate more closely with other researchers for even more robust biological validation of these results. In this regard, we present a summarized version of the future scope of this thesis.

We have developed PyPredT6, in Chapter 3, which offers users to check whether a protein is a putative T6 effector or not. Inclusion of evolutionary information-based features for identification of T6 effectors may improve the accuracy of these predictions. More features can be included, which would enhance the predictive performance of PyPredT6. Effector proteins are being discovered every day, which would eventually result in more data leading to more accurate prediction of T6 effectors. A detailed biological validation for each putative predicted T6 effector proteins is essential, which forms the scope for further study. The methodology can be extended to other pathogens, whose genomes and proteomes are either partially or fully mapped.

In Chapter 4, the oversampling algorithm CQNR and effector protein predictor EPP3D have been developed. We believe that more 3D structure-based features can be extracted in future from newly discovered effector proteins. Thus we would incorporate more features in future based on 3D structure of effector proteins along with the existing ones to develop a more robust classifier. With the advancement in machine learning methodology, a better and more sophisticated classification technique can be used to develop EPP3D. As more and more new secretion systems are being discovered, more types of effector proteins can be included for designing a more versatile classifier.

Apart from the prediction of T3, T4, T6 and T7 effector proteins, identification of the toxins liberated by T1, T2 and T5 secretion systems, is crucial. *In silico* identification of these toxins would be facilitated by the discovery of more experimentally validated toxins. Development of robust systems for identification of effector proteins considering their primary, secondary and tertiary structures would facilitate efficient toxin identification.

In Chapter 5, the training set for DeepT7 contained a limited number of T7 effector proteins. With improvements in biological experimentation, it is expected that more effector proteins will be discovered. More the number of training samples, more accurate will be the performance of the classifier. Along with proteins, more features such as tertiary structure and quaternary structure-based features, might get discovered. This would lead to the development of a more accurate prediction system.

The effect of toxins have been explored only on glycolysis and TCA pathway cycle in Chapter 6. We found 52 toxins from KEGG. In the future, more toxins will be discovered. The effect of more toxins need to be considered further to study their effect on all the pathways from KEGG and other databases. We have explored the effect of one toxin at a time on glycolysis and TCA pathways individually. The simultaneous effect of more than one toxin on these integrated pathways may lead to substantially novel discoveries, to give more insight into pathogen dynamics. Moreover, only KEGG database has been used for ASAPP. Other metabolic pathway databases should also be taken into consideration.

We have assumed, in Chapter 7, the interactions to be undirected. Direction of the interactions among the proteins is an important issue of the signal transduction pathways. We would like to enhance the representation of biological networks by assigning direction to interactions of the network so that cause-effect relationships can be analyzed better. With time, we hope the nature of the filtered interactions can be derived and validated experimentally. This will help in analyzing dynamics of the entire pathway system in the host body without limiting to selected ones.

A host cell function is manifested by an integrated and coordinated activity of gene regulatory, metabolic and signal transduction pathways. These pathways do not work in isolation. Their functionality and efficacy depend on each other. Therefore, there is a dire need to study the effect of such toxins on integrated pathways of the host. The interactions among all possible pathogens and their hosts need to be examined in detail. The study needs to be extended for viruses as well. Given the number of fatal diseases, like COVID-19, Ebola and AIDS, due to viral infection, the effect of the proteins liberated by these viruses on host pathways need to be investigated.

Identification of perturbing agents and their effect on pathways forms a study in the domain of host-pathogen interactions, which largely depends on the fields of feature extraction, classification and pathway prediction. Thus, it is evident that progress in these fields of study would have a massive impact on this domain. Since the feature set plays a vital role in identification of toxins, the improvement in feature extraction techniques will enhance the identification. More sophisticated feature extraction methodology would lead to a more potent feature set, encompassing a wider variety of features. Apart from feature extraction, classification is an important aspect of effector identification. As previously mentioned, with development of advanced classification techniques, the prediction accuracy of effector identification systems would improve. Pathway prediction forms the basis of analyzing the effect of toxins on host pathways. Prediction of pathways, considering metabolites and enzymes, would appropriately reflect the actual biological scenario.

Despite widespread advances in medical science, infectious diseases continue to have devastating consequences for human population in many parts of the world. Although the incidence of many infectious diseases in the world has decreased due to the introduction of various vaccinations, annual resurgences continue even though children have been vaccinated by the age of school entry. The fact that new pathogens are getting discovered every day indicates emergence of new diseases being a regular phenomenon. The sudden occurrence of the recent pandemic COVID-19 proves the same. In this regard, we believe that the domain of *in silico* prediction of toxins and their effect on host pathways based on feature extraction, classification and pathway prediction, awaits substantial exploration.

Appendix A

Supporting Information

A.1 Chapter 3

Table A.1: CPU time analysis of PyPredT6. The column “Sequence count” depicts the number of nucleotide and amino acid sequences in each of the random set of sequences whose classes are to be predicted. Here, a single sequence refers to a pair of nucleotide and the corresponding amino acid sequences. The column “Feature extraction time” indicates the time required by PyPredT6 to extract the features from the sequences. The column “Feature extraction rate” depicts the time needed to extract features from a single sequence. The column “Training time” denotes the time required for training PyPredT6. The column “Total time” is the sum of T_E and T_T. Averages of total time (T_S) and feature extraction time (T_E) over a varying number of sequences are not comparable. Hence these averages have been marked as “NA” (not applicable).

	Sequence count (n)	Feature extraction time (T_E in sec)	Feature extraction rate ($\frac{T_E}{n}$ in sec/sequence)	Training time (T_T in sec)	Total time ($T_S = T_E + T_T$ in sec)
	Random set 1-10	0.4481	0.0448	313.7346	314.1827
	Random set 2-20	0.8678	0.0433	316.6949	317.5627
	Random set 3-30	4.118	0.1372	313.4174	317.5354
Average	NA	NA	**0.0751**	**314.61**	NA

A.2 Chapter 4

A.2.1 Analysis of combination of cluster validity index and clustering algorithm

It has been analyzed which combination of clustering algorithm-validity index works best [154]. We have tabulated the results of the analysis in Table A.2, where each cell holds the number of times the respective validity index designated by the row and the clustering algorithm designated by the column, has given a good performances. It is clearly seen that the combination of K-means and Davis-Bouldin index has given the best performance for most of the dataset compared to the other validity indices.

Table A.2: Comparison of performance of different clustering techniques with different cluster validity indices on several datasets

Cluster validity Index	K-Mean	Pam	Fuzzy
Dunn index	5	3	2
Davis-Bouldin index	10	9	4
Silhouette index	4	1	2
C-index	5	5	1
Goodman-Kruskal index	4	2	1
Isolation index	5	7	1
Partition coefficient index	0	0	0
Classification entropy index	4	2	2
Partition index	1	0	1
Separation index	3	1	2
Xie and Beni's index	1	1	4
Fukuyama and Sugeno index	2	0	2
Fuzzy hypervolume index	1	2	2
Alternative dunn index	0	1	0
Dave's modification of the PC index	3	1	3
Partition coefficient and exponential separation index	0	1	3
Index based on Akaikes in formation criterion	1	2	4
Compose within and Between scattering index	5	1	5
PBMF-index	2	0	7

A.2.2 Results

In this section, we provide an elaborate comparison of the various oversampling algorithms in a tabulated format. Tables A.3 to A.7 tabulate the performance comparison of CQNR with other oversampling techniques based on *Accuracy, Sensitivity, Specificity, F-score* and *G-mean* respectively. An elaborate tabulation of the performance of EPP3D over various subsets of effector proteins has been provided in the Tables A.8 to A.10 based on *Accuracy*, κ score and *MCC* respectively. Three independent datasets consisting of T3, T4 and T6 effector proteins have been created to analyze the performance of EPP3D. The results have been tabulated in the Tables A.11 to A.13.

Table A.3: Comparison of CQNR with other over-sampling algorithms on various datasets with respect to *Accuracy*.

Dataset	Method	SVM	MLP	NB	kNN	DT
Pima Diabetes	CQNR	**0.8152**	0.7666	0.7369	0.7901	0.7504
	Unbalanced	0.7565	0.6994	0.6970	0.7261	0.6875
	Random over-sampling	0.7685	0.6750	0.6687	0.7354	0.6956
	SMOTE	0.7693	0.6698	0.6765	0.7437	0.7036
	Borderline-SMOTE	0.7534	0.6876	0.6845	0.7352	0.6925
	C-SMOTE	0.7532	0.6723	0.6934	0.7266	0.6995
	Safe-level-SMOTE	0.8034	0.7798	0.7264	0.7842	0.7432
Haberman	CQNR	0.7349	0.7181	0.6916	**0.7474**	0.7046
	Unbalanced	0.7427	0.6089	0.7295	0.6464	0.6625
	Random over-sampling	0.7314	0.5867	0.6945	0.6498	0.6712
	SMOTE	0.7498	0.5945	0.7054	0.6545	0.6834
	Borderline-SMOTE	0.7446	0.6724	0.7367	0.6934	0.6743
	C-SMOTE	0.7245	0.6638	0.7323	0.6865	0.6834
	Safe-level-SMOTE	0.7267	0.7034	0.7143	0.6932	0.6832
Spambase	CQNR	**0.9448**	0.9321	0.9217	0.9254	0.9238
	Unbalanced	0.9272	0.9387	0.8809	0.9031	0.8955
	Random over-sampling	0.9297	0.9156	0.8954	0.9098	0.9045
	SMOTE	0.9356	0.9084	0.8743	0.8954	0.9032
	Borderline-SMOTE	0.9254	0.9196	0.8842	0.9145	0.9032
	C-SMOTE	0.9165	0.9154	0.8902	0.9265	0.9174
	Safe-level-SMOTE	0.9034	0.9254	0.9162	0.9147	0.9165
Hill-Valley	CQNR	0.5596	**0.6856**	0.5270	0.5574	0.5955
	Unbalanced	0.5127	0.6763	0.5382	0.5602	0.5730
	Random over-sampling	0.5267	0.6687	0.5395	0.5865	0.5798
	SMOTE	0.5156	0.6743	0.5476	0.5732	0.5953
	Borderline-SMOTE	0.5386	0.6734	0.5498	0.5534	0.5834
	C-SMOTE	0.5478	0.6832	0.5585	0.5397	0.5643
	Safe-level-SMOTE	0.5387	0.6623	0.5390	0.5496	0.5853
Blood transfusion	CQNR	**0.7887**	0.7824	0.6481	0.7767	0.7692
	Unbalanced	0.7663	0.7044	0.7461	0.7308	0.7170
	Random over-sampling	0.7645	0.7012	0.7576	0.7476	0.7246
	SMOTE	0.7698	0.6943	0.7698	0.7534	0.7032
	Borderline-SMOTE	0.7745	0.7137	0.7632	0.7477	0.7145
	C-SMOTE	0.7634	0.7253	0.7742	0.7597	0.7254
	Safe-level-SMOTE	0.7596	0.7723	0.6365	0.7432	0.7498
Synthetic dataset 1	CQNR	0.4893	0.4824	0.3481	0.4767	0.4692
	Unbalanced	**0.7663**	0.7044	0.7461	0.7308	0.7170
	Random over-sampling	0.4645	0.40.12	0.4576	0.4476	0.4246
	SMOTE	0.46.98	0.4943	0.4698	0.4534	0.4032
	Borderline-SMOTE	0.4745	0.4137	0.4632	0. 4477	0.4145
	C-SMOTE	0.4634	0.4253	0.4742	0.4597	0.4254
	Safe-level-SMOTE	0.4596	0.4723	0.3365	0.4432	0.4498
Synthetic dataset 2	CQNR	0.3683	0.3792	0.3565	0.3572	0.3609
	Unbalanced	0.8436	**0.8742**	0.8593	0.8245	0.8364
	Random over-sampling	0.3273	0.3062	0.3162	0.3363	0.3424
	SMOTE	0.3523	0.3274	0.3127	0.3328	0.3532
	Borderline-SMOTE	0.3423	0.3734	0.3612	0.3376	0.3447
	C-SMOTE	0.3382	0.3474	0.3115	0.3181	0.3294
	Safe-level-SMOTE	0.3571	0.3483	0.3173	0.3627	0.3407
Synthetic dataset 3	CQNR	0.3374	0.3565	0.3493	0.3655	0.3462
	Unbalanced	**0.8963**	0.8844	0.8661	0.8908	0.8770
	Random over-sampling	0.2905	0.3034	0.2859	0.3071	0.2704
	SMOTE	0.2762	0.2949	0.3193	0.3275	0.3010
	Borderline-SMOTE	0.3174	0.3496	0.3147	0.3265	0.3249
	C-SMOTE	0.2854	0.2798	0.2864	0.2563	0.2642
	Safe-level-SMOTE	0.2639	0.2858	0.2934	0.2546	0.2759

Table A.4: Comparison of CQNR with other over-sampling algorithms on various datasets with respect to *Sensitivity*.

Dataset	Method	SVM	MLP	NB	kNN	DT
Pima Diabetes	CQNR	0.7721	0.7215	0.7044	0.5625	0.7854
	Unbalanced	0.7223	0.6861	0.7585	0.4877	0.7749
	Random over-sampling	0.7460	0.6741	0.7984	0.5703	0.7001
	SMOTE	0.7718	0.7151	0.6043	0.5932	0.7941
	Borderline-SMOTE	0.7793	0.7306	0.6643	0.6038	0.7427
	C-SMOTE	0.7529	0.7293	0.7323	0.6042	0.7277
	Safe-level-SMOTE	0.7328	0.6387	0.7034	0.6237	0.7306
Haberman	CQNR	0.7019	0.5060	0.5891	0.0274	0.6796
	Unbalanced	0.4295	0.3599	0.5076	0.5304	0.6725
	Random over-sampling	0.7497	0.5303	0.5733	0.3065	0.6233
	SMOTE	0.6406	0.5056	0.6370	0.4831	0.5947
	Borderline-SMOTE	0.6545	0.5614	0.6804	0.4624	0.7298
	C-SMOTE	0.6188	0.6116	0.5867	0.3440	0.7022
	Safe-level-SMOTE	0.6974	0.5333	0.5771	0.5131	0.6560
Spambase	CQNR	0.9248	0.9295	0.8976	0.3298	0.8913
	Unbalanced	0.9273	0.9000	0.9299	0.2429	0.8959
	Random over-sampling	0.9559	0.9118	0.7882	0.3163	0.9051
	SMOTE	0.9287	0.9400	0.9689	0.3111	0.8877
	Borderline-SMOTE	0.9124	0.9968	0.9078	0.3012	0.9026
	C-SMOTE	0.9349	0.9050	0.8467	0.2825	0.8536
	Safe-level-SMOTE	0.9282	0.8938	0.9120	0.2745	0.8732
Hill-Valley	CQNR	0.3416	0.5042	0.4235	0.2979	0.5188
	Unbalanced	0.3252	0.3328	0.4498	0.2039	0.6108
	Random over-sampling	0.3181	0.3576	0.3798	0.2633	0.5853
	SMOTE	0.3717	0.4657	0.4133	0.2530	0.5416
	Borderline-SMOTE	0.3400	0.5534	0.4467	0.3483	0.6662
	C-SMOTE	0.4039	0.5363	0.4087	0.2976	0.5500
	Safe-level-SMOTE	0.3992	0.5546	0.3890	0.3205	0.5844
Blood transfusion	CQNR	0.7022	0.6811	0.7591	0.7145	0.6083
	Unbalanced	0.4213	0.5793	0.7625	0.6258	0.6749
	Random over-sampling	0.4043	0.6154	0.5906	0.6518	0.6918
	SMOTE	0.4894	0.6273	0.6501	0.6144	0.5906
	Borderline-SMOTE	0.6328	0.5963	0.6927	0.6397	0.6113
	C-SMOTE	0.5991	0.6493	0.7397	0.6125	0.6036
	Safe-level-SMOTE	0.5759	0.6342	0.7030	0.7553	0.6168
Synthetic dataset 1	CQNR	0.8919	0.8661	0.6054	0.7917	0.9569
	Unbalanced	0.6964	0.8156	0.7460	0.8351	0.8564
	Random over-sampling	0.7378	0.8163	0.7550	0.7816	0.8551
	SMOTE	0.9674	0.8512	0.6966	0.8561	0.9369
	Borderline-SMOTE	0.8678	0.8402	0.6215	0.8126	0.8964
	C-SMOTE	0.9124	0.8357	0.7290	0.7903	0.9176
	Safe-level-SMOTE	0.9667	0.7666	0.6398	0.7770	0.9271
Synthetic dataset 2	CQNR	0.9929	0.9063	0.7688	0.9693	0.9805
	Unbalanced	0.9428	0.8990	0.9114	0.9296	0.8931
	Random over-sampling	0.9654	0.9134	0.7318	0.8959	0.9713
	SMOTE	0.9321	0.9177	0.8520	0.9368	0.9387
	Borderline-SMOTE	0.9274	0.9277	0.7547	0.9071	0.9723
	C-SMOTE	0.9935	0.9010	0.8939	0.9418	0.9427
	Safe-level-SMOTE	0.9977	0.9160	0.7650	0.9686	0.9825
Synthetic dataset 3	CQNR	0.8610	0.7328	0.5844	0.8030	0.8907
	Unbalanced	0.7513	0.7257	0.7587	0.7728	0.7939
	Random over-sampling	0.7829	0.7345	0.6100	0.7365	0.8264
	SMOTE	0.8616	0.7355	0.7750	0.8072	0.8439
	Borderline-SMOTE	0.8748	0.7424	0.5574	0.8319	0.8970
	C-SMOTE	0.8049	0.7562	0.5625	0.7985	0.8684
	Safe-level-SMOTE	0.8910	0.7296	0.4456	0.7923	0.8836

Table A.5: Comparison of CQNR with other over-sampling algorithms on various datasets with respect to *Specificity*.

Dataset	Method	SVM	MLP	NB	kNN	DT
Pima Diabetes	CQNR	0.7934	0.7492	0.7205	0.6513	0.7193
	Unbalanced	0.6732	0.6135	0.6028	0.5723	0.6194
	Random over-sampling	0.6284	0.6924	0.6024	0.5974	0.7146
	SMOTE	0.7028	0.6724	0.6493	0.5523	0.7085
	Borderline-SMOTE	0.7284	0.7185	0.6354	0.5834	0.7254
	C-SMOTE	0.7187	0.7037	0.6398	0.5246	0.7311
	Safe-level-SMOTE	0.7824	0.6183	0.7042	0.5734	0.7023
Haberman	CQNR	0.7146	0.6954	0.6089	0.5194	0.6884
	Unbalanced	0.3985	0.3837	0.5284	0.5492	0.5183
	Random over-sampling	0.5385	0.5046	0.5735	0.4824	0.5937
	SMOTE	0.6184	0.5735	0.5352	0.5732	0.6257
	Borderline-SMOTE	0.6426	0.5853	0.5245	0.4632	0.6643
	C-SMOTE	0.6524	0.5632	0.5643	0.3692	0.6653
	Safe-level-SMOTE	0.6742	0.5257	0.5723	0.5631	0.6632
Spambase	CQNR	0.9624	0.9593	0.9193	0.2725	0.9193
	Unbalanced	0.9183	0.9532	0.8432	0.3843	0.8493
	Random over-sampling	0.9193	0.9232	0.8392	0.3038	0.8624
	SMOTE	0.9064	0.9150	0.8034	0.2862	0.8724
	Borderline-SMOTE	0.9046	0.9194	0.8814	0.2725	0.8825
	C-SMOTE	0.9028	0.8823	0.8792	0.2763	0.8256
	Safe-level-SMOTE	0.9073	0.8735	0.8836	0.2936	0.8917
Hill-Valley	CQNR	0.3716	0.4873	0.4364	0.3625	0.5763
	Unbalanced	0.3192	0.3027	0.3437	0.2523	0.5025
	Random over-sampling	0.3623	0.3018	0.4293	0.2193	0.5153
	SMOTE	0.3273	0.3193	0.4254	0.2826	0.5173
	Borderline-SMOTE	0.3193	0.4326	0.4103	0.2352	0.5018
	C-SMOTE	0.3002	0.4163	0.4283	0.3192	0.5017
	Safe-level-SMOTE	0.3062	0.4017	0.4263	0.2083	0.5012
Blood transfusion	CQNR	0.6153	0.6004	0.6156	0.6725	0.6173
	Unbalanced	0.3183	0.5028	0.5027	0.5192	0.5093
	Random over-sampling	0.4103	0.5536	0.6193	0.5204	0.5174
	SMOTE	0.5184	0.5274	0.6074	0.5834	0.6173
	Borderline-SMOTE	0.4924	0.6132	0.5902	0.6173	0.5836
	C-SMOTE	0.6187	0.6024	0.5080	0.6264	0.6132
	Safe-level-SMOTE	0.5037	0.6028	0.6183	0.6254	0.6037
Synthetic dataset 1	CQNR	0.9153	0.7946	0.5342	0.8153	0.9013
	Unbalanced	0.7163	0.7726	0.7621	0.8172	0.8073
	Random over-sampling	0.7013	0.7935	0.7163	0.6934	0.8012
	SMOTE	0.6212	0.8023	0.5139	0.8193	0.8734
	Borderline-SMOTE	0.8137	0.7935	0.5934	0.7823	0.8783
	C-SMOTE	0.8693	0.8013	0.6230	0.8192	0.8304
	Safe-level-SMOTE	0.8183	0.8426	0.6794	0.7937	0.7894
Synthetic dataset 2	CQNR	0.9323	0.9264	0.7497	0.9429	0.9636
	Unbalanced	0.9182	0.8845	0.9623	0.9012	0.9154
	Random over-sampling	0.9293	0.8723	0.7453	0.9193	0.9047
	SMOTE	0.8934	0.8734	0.8016	0.9002	0.9182
	Borderline-SMOTE	0.9263	0.8683	0.7193	0.9173	0.9132
	C-SMOTE	0.9013	0.9312	0.8016	0.9132	0.9034
	Safe-level-SMOTE	0.9183	0.8735	0.7916	0.9265	0.9134
Synthetic dataset 3	CQNR	0.8753	0.7442	0.5788	0.7845	0.9023
	Unbalanced	0.7012	0.7132	0.7386	0.7017	0.6948
	Random over-sampling	0.6983	0.7025	0.5274	0.7623	0.7163
	SMOTE	0.7192	0.7037	0.6023	0.7142	0.7274
	Borderline-SMOTE	0.8012	0.7182	0.5028	0.6395	0.8726
	C-SMOTE	0.7163	0.7037	0.6342	0.7715	0.8273
	Safe-level-SMOTE	0.8183	0.7074	0.6524	0.7258	0.8183

Table A.6: Comparison of CQNR with other over-sampling algorithms on various datasets with respect to *F-score*.

Dataset	Method	SVM	MLP	NB	kNN	DT
Pima Diabetes	CQNR	0.7844	**0.8463**	0.7003	0.6325	0.7756
	Unbalanced	0.6025	0.8079	0.5862	0.4711	0.6035
	Random over-sampling	0.6154	0.8356	0.6321	0.5683	0.6395
	SMOTE	0.6996	0.8274	0.6142	0.5294	0.6932
	Borderline-SMOTE	0.7347	0.8395	0.6043	0.6143	0.7563
	C-SMOTE	0.7664	0.8153	0.6794	0.5936	0.7194
	Safe-level-SMOTE	0.7536	0.8265	0.6964	0.6493	0.7377
Haberman	CQNR	0.7295	0.5398	0.7032	0.6532	0.7195
	Unbalanced	**0.7474**	0.2222	0.7375	0.7051	0.7777
	Random over-sampling	0.7264	0.3284	0.7293	0.6592	0.6364
	SMOTE	0.7142	0.4583	0.7143	0.6834	0.6945
	Borderline-SMOTE	0.7385	0.3965	0.7134	0.6156	0.6394
	C-SMOTE	0.7133	0.3825	0.7285	0.6262	0.6283
	Safe-level-SMOTE	0.7342	0.3954	0.7325	0.6954	0.6834
Spambase	CQNR	0.9427	**0.9574**	0.907	0.6828	0.8835
	Unbalanced	0.9095	0.9434	0.8642	0.5832	0.8494
	Random over-sampling	0.9134	0.9254	0.8342	0.5932	0.7936
	SMOTE	0.9364	0.9173	0.8721	0.6731	0.7836
	Borderline-SMOTE	0.9255	0.9472	0.8854	0.6384	0.8557
	C-SMOTE	0.9174	0.9374	0.8932	0.6693	0.8644
	Safe-level-SMOTE	0.9264	0.9352	0.8872	0.6573	0.8562
Hill-Valley	CQNR	0.6595	0.6395	0.6283	0.6679	0.6282
	Unbalanced	0.6398	0.1834	0.6509	0.6693	0.4984
	Random over-sampling	0.6186	0.4587	0.6283	0.6492	0.5832
	SMOTE	0.6645	0.5686	0.6743	0.6382	0.5734
	Borderline-SMOTE	0.6248	0.5823	0.6194	0.6599	0.5986
	C-SMOTE	0.6385	0.6169	**0.6836**	0.6749	0.6073
	Safe-level-SMOTE	0.6277	0.5973	0.6294	0.6593	0.6129
Blood transfusion	CQNR	0.6909	0.6723	0.6956	0.6846	**0.6993**
	Unbalanced	0.2318	0.6772	0.4603	0.4000	0.4142
	Random over-sampling	0.5673	0.6749	0.5493	0.5833	0.5837
	SMOTE	0.5985	0.6382	0.6832	0.5867	0.5737
	Borderline-SMOTE	0.6892	0.6947	0.5938	0.6365	0.6835
	C-SMOTE	0.5574	0.6321	0.6294	0.6839	0.5982
	Safe-level-SMOTE	0.5757	0.5963	0.6732	0.6245	0.6596
Synthetic dataset 1	CQNR	0.9103	0.8943	0.5978	0.7392	**0.9266**
	Unbalanced	0.5517	0.8963	0.5396	0.6397	0.7428
	Random over-sampling	0.8504	0.8274	0.5839	0.6491	0.8355
	SMOTE	0.8375	0.8593	0.5583	0.6859	0.8296
	Borderline-SMOTE	0.8466	0.9042	0.5501	0.7305	0.9175
	C-SMOTE	0.9063	0.8395	0.5698	0.7195	0.8674
	Safe-level-SMOTE	0.8739	0.9174	0.5497	0.6855	0.8947
Synthetic dataset 2	CQNR	0.9597	0.8714	0.7992	0.8360	**0.9708**
	Unbalanced	0.9152	0.8698	0.7573	0.8145	0.8500
	Random over-sampling	0.9274	0.8593	0.7854	0.8164	0.8963
	SMOTE	0.9478	0.8947	0.7793	0.8148	0.9477
	Borderline-SMOTE	0.9164	0.8863	0.7586	0.8264	0.8567
	C-SMOTE	0.9116	0.8399	0.7605	0.8550	0.9588
	Safe-level-SMOTE	0.9289	0.8457	0.7688	0.8465	0.9356
Synthetic dataset 3	CQNR	0.8893	0.7502	0.5816	0.7342	0.9021
	Unbalanced	0.8402	0.9205	0.6923	0.7497	0.6101
	Random over-sampling	0.8465	0.8473	0.5938	0.7194	0.8574
	SMOTE	0.8946	**0.9384**	0.6284	0.7395	0.8475
	Borderline-SMOTE	0.8765	0.7284	0.6948	0.7276	0.8375
	C-SMOTE	0.8594	0.7956	0.5867	0.749	0.8974
	Safe-level-SMOTE	0.8673	0.8475	0.5734	0.7150	0.8975

Table A.7: Comparison of CQNR with other over-sampling algorithms on various datasets with respect to *G-mean*.

Dataset	Method	SVM	MLP	NB	kNN	DT
Pima Diabetes	CQNR	**0.7827**	0.7352	0.7124	0.6053	0.7516
	Unbalanced	0.6973	0.6488	0.6762	0.5283	0.6928
	Random over-sampling	0.6847	0.6832	0.6935	0.5837	0.7073
	SMOTE	0.7365	0.6934	0.6264	0.5724	0.7501
	Borderline-SMOTE	0.7534	0.7245	0.6497	0.5935	0.7340
	C-SMOTE	0.7356	0.7164	0.6845	0.5630	0.7294
	Safe-level-SMOTE	0.7572	0.6284	0.7038	0.5980	0.7163
Haberman	CQNR	**0.7082**	0.5932	0.5989	0.1194	0.6840
	Unbalanced	0.4137	0.3716	0.5179	0.5397	0.5904
	Random over-sampling	0.6354	0.5173	0.5734	0.3845	0.6083
	SMOTE	0.6294	0.5385	0.5839	0.5262	0.6100
	Borderline-SMOTE	0.6485	0.5732	0.5974	0.4628	0.6963
	C-SMOTE	0.6354	0.5869	0.5754	0.3564	0.6835
	Safe-level-SMOTE	0.6857	0.5295	0.5747	0.5375	0.6596
Spambase	CQNR	0.9434	0.9443	0.9084	0.2998	0.9052
	Unbalanced	0.9228	0.9262	0.8855	0.3055	0.8723
	Random over-sampling	0.9374	0.9175	0.8133	0.3100	0.8835
	SMOTE	0.9175	0.9274	0.8823	0.2984	0.8800
	Borderline-SMOTE	0.9085	**0.9573**	0.8945	0.2865	0.8925
	C-SMOTE	0.9187	0.8936	0.8628	0.2794	0.8395
	Safe-level-SMOTE	0.9177	0.8836	0.8977	0.2839	0.8824
Hill-Valley	CQNR	0.3563	0.4957	0.4299	0.3286	0.5468
	Unbalanced	0.3222	0.3174	0.3932	0.2268	0.5540
	Random over-sampling	0.3395	0.3285	0.4038	0.2403	0.5492
	SMOTE	0.3488	0.3856	0.4193	0.2674	0.5293
	Borderline-SMOTE	0.3295	0.4893	0.4281	0.2862	**0.5782**
	C-SMOTE	0.3482	0.4725	0.4184	0.3082	0.5253
	Safe-level-SMOTE	0.3496	0.4720	0.4072	0.2584	0.5412
Blood transfusion	CQNR	0.6573	0.6395	0.6836	**0.6932**	0.6128
	Unbalanced	0.3662	0.5397	0.6191	0.5700	0.5863
	Random over-sampling	0.4073	0.5837	0.6048	0.5824	0.5983
	SMOTE	0.5037	0.5752	0.6284	0.5987	0.6038
	Borderline-SMOTE	0.5582	0.6047	0.6394	0.6284	0.5973
	C-SMOTE	0.6088	0.6254	0.6130	0.6194	0.6084
	Safe-level-SMOTE	0.5386	0.6183	0.6593	0.6873	0.6102
Synthetic dataset 1	CQNR	0.9035	0.8296	0.5687	0.8034	**0.9287**
	Unbalanced	0.7063	0.7938	0.7540	0.8261	0.8315
	Random over-sampling	0.7193	0.8048	0.7354	0.7362	0.8277
	SMOTE	0.7985	0.8264	0.5983	0.8375	0.9046
	Borderline-SMOTE	0.8403	0.8165	0.6073	0.7973	0.8873
	C-SMOTE	0.8906	0.8183	0.6739	0.8046	0.8729
	Safe-level-SMOTE	0.8894	0.8037	0.6593	0.7853	0.8937
Synthetic dataset 2	CQNR	0.9621	0.9163	0.7592	0.9560	**0.9720**
	Unbalanced	0.9304	0.8917	0.9365	0.9153	0.9042
	Random over-sampling	0.9472	0.8926	0.7385	0.9075	0.9374
	SMOTE	0.9465	0.8953	0.8264	0.9183	0.9284
	Borderline-SMOTE	0.9642	0.8975	0.7368	0.9122	0.9673
	C-SMOTE	0.9463	0.9160	0.8465	0.9274	0.9538
	Safe-level-SMOTE	0.9572	0.8945	0.7782	0.9473	0.9473
Synthetic dataset 3	CQNR	0.8681	0.7385	0.5816	0.7937	**0.8965**
	Unbalanced	0.7258	0.7194	0.7486	0.7364	0.7427
	Random over-sampling	0.7394	0.7183	0.5672	0.7493	0.7694
	SMOTE	0.7872	0.7194	0.6832	0.7593	0.7835
	Borderline-SMOTE	0.8372	0.7302	0.5294	0.7294	0.8847
	C-SMOTE	0.7593	0.7295	0.5973	0.7849	0.8476
	Safe-level-SMOTE	0.8539	0.7184	0.5392	0.7583	0.8503

Table A.8: Summary of *Accuracy* of the classifiers on the experimentally verified pathogenic effector proteins after 20 fold cross-validation before and after dataset balancing. '+' indicates the classes merged into a single class. '/' indicates that the classes on either side of '/' are treated as a separate class.

Effector Protein set	Method	SVM	MLP	NB	kNN	RF	Consensus
T3/T4/T6 (3 class)	CQNR	0.5932	0.6085	0.5964	0.5896	0.5734	**0.6432**
	Unbalanced	0.5534	0.5467	0.5301	0.5376	0.4814	0.5634
	Random oversampling	0.5467	0.5583	0.5673	0.5284	0.5034	0.5539
	SMOTE	0.5527	0.5639	0.5595	0.5467	0.5273	0.5686
	Borderline-SMOTE	0.5473	0.5783	0.5247	0.5536	0.5139	0.5542
	C-SMOTE	0.5636	0.5428	0.5385	0.5312	0.5467	0.5724
	Safe-level-SMOTE	0.5542	0.5547	0.5467	0.5467	0.5305	0.5932
T3/(T4+T6) (2 class)	CQNR	0.6708	0.6835	0.7258	0.673	0.7019	**0.7643**
	Unbalanced	0.5851	0.4565	0.5071	0.6428	0.5091	0.5943
	Random oversampling	0.5943	0.4626	0.5283	0.6183	0.5193	0.6081
	SMOTE	0.6081	0.4793	0.5832	0.6264	0.5247	0.6372
	Borderline-SMOTE	0.6372	0.6174	0.6273	0.6023	0.5485	0.6835
	C-SMOTE	0.6153	0.5802	0.6193	0.6503	0.5924	0.6503
	Safe-level-SMOTE	0.6005	0.5573	0.6593	0.6242	0.6284	0.6372
T4/(T3+T6) (2 class)	CQNR	0.5866	0.5786	0.5650	0.5969	0.5317	**0.6587**
	Unbalanced	0.5308	0.5217	0.5217	0.5016	0.4666	0.5582
	Random oversampling	0.5293	0.5300	0.5284	0.5172	0.4738	0.5386
	SMOTE	0.5458	0.5493	0.5382	0.5269	0.4823	0.5683
	Borderline-SMOTE	0.5572	0.5577	0.5423	0.5307	0.4902	0.5969
	C-SMOTE	0.5461	0.5247	0.5529	0.5537	0.5022	0.5786
	Safe-level-SMOTE	0.5582	0.5683	0.5386	0.5264	0.5137	0.5866
T6/(T3+T4) (2 class)	CQNR	0.7975	0.7632	0.7925	0.6717	0.7613	**0.8425**
	Unbalanced	0.7432	0.7173	0.5214	0.6204	0.6183	0.7694
	Random oversampling	0.7523	0.7329	0.5348	0.6395	0.6283	0.7394
	SMOTE	0.7694	0.7284	0.5492	0.6439	0.6328	0.7845
	Borderline-SMOTE	0.7721	0.7394	0.5571	0.6538	0.642	0.7845
	C-SMOTE	0.7845	0.7536	0.5832	0.6692	0.6529	0.8135
	Safe-level-SMOTE	0.7845	0.7496	0.6135	0.6799	0.6802	0.7975
T3/T4 (2 class)	CQNR	0.6975	0.6766	0.6312	0.5841	0.6483	**0.7286**
	Unbalanced	0.6150	0.5135	0.5550	0.5294	0.5025	0.6376
	Random oversampling	0.6293	0.5239	0.5623	0.5328	0.5103	0.6432
	SMOTE	0.6439	0.5356	0.5703	0.5495	0.5246	0.6263
	Borderline-SMOTE	0.6304	0.5406	0.5821	0.5574	0.5397	0.6432
	C-SMOTE	0.6534	0.5912	0.6134	0.5603	0.5426	0.6694
	Safe-level-SMOTE	0.6604	0.6234	0.6823	0.5782	0.5583	0.7032
T3/T6 (2 class)	CQNR	0.5791	0.5576	0.5433	0.6009	**0.6918**	0.6557
	Unbalanced	0.5250	0.5165	0.5450	0.5564	0.5238	0.5424
	Random oversampling	0.5129	0.5239	0.5120	0.5632	0.5329	0.5238
	SMOTE	0.5353	0.5320	0.4932	0.5796	0.5495	0.5463
	Borderline-SMOTE	0.5403	0.5473	0.5053	0.5932	0.5502	0.5328
	C-SMOTE	0.5587	0.5320	0.5249	0.5823	0.5675	0.5638
	Safe-level-SMOTE	0.5435	0.5534	0.5129	0.5238	0.5758	0.5638
T4/T6 (2 class)	CQNR	0.6950	0.7070	0.7025	0.5752	0.6521	**0.7494**
	Unbalanced	0.6663	0.6470	0.5550	0.5308	0.6266	0.6754
	Random oversampling	0.6734	0.6534	0.5630	0.5403	0.6302	0.6832
	SMOTE	0.6839	0.6604	0.5734	0.5583	0.6439	0.6932
	Borderline-SMOTE	0.6539	0.6723	0.5934	0.5245	0.6530	0.6635
	C-SMOTE	0.6634	0.6534	0.5834	0.5139	0.6639	0.6721
	Safe-level-SMOTE	0.6534	0.6234	0.6329	0.5402	0.6724	0.6723
T3/T4/T6/Other/Non-effector (5 class)	CQNR	0.6394	0.7553	0.6556	0.6474	0.6635	**0.8543**
	Unbalanced	0.6897	0.6757	0.6143	0.6634	0.6045	0.6943
	Random oversampling	0.6923	0.6823	0.6193	0.6739	0.6138	0.7024
	SMOTE	0.6002	0.6946	0.6234	0.6804	0.6482	0.7394
	Borderline-SMOTE	0.6943	0.7023	0.6804	0.6923	0.6397	0.7556
	C-SMOTE	0.6823	0.6736	0.6239	0.7045	0.7039	0.6823
	Safe-level-SMOTE	0.6024	0.7020	0.6329	0.6624	0.6503	0.6946
T3/Non-effectors (2 class)	CQNR	0.6527	0.7361	0.5861	0.6944	0.7083	**0.7865**
	Unbalanced	0.5250	0.5165	0.5367	0.5564	0.5238	0.5424
	Random oversampling	0.5424	0.5000	0.5250	0.5000	0.5165	0.5861
	SMOTE	0.5000	0.5424	0.5424	0.5861	0.5424	0.5238
	Borderline-SMOTE	0.5165	0.5861	0.5734	0.5734	0.5165	0.6527
	C-SMOTE	0.5734	0.5238	0.5165	0.5238	0.5861	0.6944
	Safe-level-SMOTE	0.6624	0.6823	0.5424	0.7427	0.6823	0.7361

Table A.8 continues: Summary of *Accuracy* of the classifiers on the experimentally verified pathogenic effector proteins after 20 fold cross-validation before and after dataset balancing. '+' indicates the classes merged into a single class. '/' indicates that the classes on either side of '/' are treated as a separate class.

Effector Protein set	Method	SVM	MLP	NB	kNN	RF	Consensus
T4/Non-effectors (2 class)	CQNR	0.6027	0.6301	0.4794	0.5205	0.6438	**0.6924**
	Unbalanced	0.5205	0.5157	0.4937	0.4825	0.4937	0.5428
	Random oversampling	0.5395	0.5638	0.5638	0.5205	0.5157	0.5924
	SMOTE	0.5428	0.5428	0.5205	0.5893	0.5205	0.6301
	Borderline-SMOTE	0.5638	0.5205	0.5893	0.5205	0.5428	0.6193
	C-SMOTE	0.6301	0.5638	0.5428	0.5638	0.5924	0.5638
	Safe-level-SMOTE	0.5893	0.5924	0.5638	0.5428	0.6301	0.6527
T6/Non-effector (2 class)	CQNR	0.7638	0.8750	0.5416	0.7777	0.8333	**0.9123**
	Unbalanced	0.6950	0.6838	0.4157	0.695	0.6838	0.7361
	Random oversampling	0.7193	0.7361	0.4923	0.6838	0.7193	0.7556
	SMOTE	0.7494	0.7694	0.4625	0.7193	0.6950	0.7694
	Borderline-SMOTE	0.7361	0.7556	0.4959	0.7494	0.7556	0.8294
	C-SMOTE	0.7694	0.6950	0.5329	0.7556	0.7494	0.8750
	Safe-level-SMOTE	0.7556	0.7494	0.5623	0.7694	0.7361	0.9023
Other/Non-effector (2 class)	CQNR	0.8472	0.9200	0.5138	0.7916	0.9166	**0.9423**
	Unbalanced	0.7827	0.7638	0.4957	0.6239	0.7827	0.8623
	Random oversampling	0.8025	0.8237	0.5658	0.6927	0.8025	0.8623
	SMOTE	0.8237	0.8025	0.5235	0.7638	0.8237	0.8853
	Borderline-SMOTE	0.8134	0.8623	0.5862	0.8623	0.8472	0.9025
	C-SMOTE	0.7827	0.8134	0.5483	0.7827	0.8134	0.932
	Safe-level-SMOTE	0.8237	0.8623	0.5638	0.8025	0.8623	0.9254
Eff/Non-effector (2 class)	CQNR	0.6800	0.5533	0.5145	0.6990	0.6116	0.7423
	Unbalanced	0.5851	0.5308	0.5048	0.5428	0.4937	0.6838
	Random oversampling	0.5943	0.5403	0.5120	0.5924	0.5638	0.7193
	SMOTE	0.6081	0.5583	0.4932	0.6301	0.5205	0.6950
	Borderline-SMOTE	0.6372	0.5245	0.5053	0.6193	0.5893	**0.7556**
	C-SMOTE	0.6153	0.5139	0.5249	0.5638	0.5428	0.7494
	Safe-level-SMOTE	0.6005	0.5402	0.5129	0.6527	0.5638	0.7361
(T3+T4+T6)/Non-effector (2 class)	CQNR	0.5393	0.5730	0.5617	0.5842	0.6741	**0.7323**
	Unbalanced	0.5308	0.5467	0.5139	0.5634	0.5428	0.6741
	Random oversampling	0.5403	0.5583	0.5245	0.5539	0.5924	0.6527
	SMOTE	0.5583	0.5639	0.5308	0.5686	0.6301	0.573
	Borderline-SMOTE	0.5245	0.5783	0.5583	0.5542	0.6193	0.6301
	C-SMOTE	0.5139	0.5428	0.5139	0.5724	0.5638	0.6193
	Safe-level-SMOTE	0.5402	0.5547	0.5403	0.5932	0.6527	0.6301
(T3+T4+T6)/Other (2 class)	CQNR	0.8181	0.9318	0.5568	0.7727	0.8636	**0.9624**
	Unbalanced	0.63013	0.6741	0.5308	0.6527	0.6741	0.7323
	Random oversampling	0.6741	0.6934	0.5403	0.6832	0.6943	0.7527
	SMOTE	0.7397	0.7248	0.5583	0.6527	0.7974	0.7627
	Borderline-SMOTE	0.7543	0.7942	0.5245	0.7241	0.8327	0.8234
	C-SMOTE	0.7248	0.7543	0.5139	0.7428	0.8837	0.8368
	Safe-level-SMOTE	0.7942	0.8364	0.5402	0.7397	0.8368	0.8837
T3/T4/T6/Other (4 class)	CQNR	0.5473	0.8368	0.5421	0.6052	0.5684	**0.8723**
	Unbalanced	0.5301	0.7527	0.5467	0.5539	0.5308	0.6741
	Random oversampling	0.5673	0.7527	0.5583	0.5686	0.5403	0.7895
	SMOTE	0.5595	0.7895	0.5639	0.5542	0.5683	0.8046
	Borderline-SMOTE	0.5247	0.7627	0.5783	0.5724	0.5245	0.7527
	C-SMOTE	0.5385	0.8046	0.5428	0.5932	0.5139	0.7627
	Safe-level-SMOTE	0.5467	0.8234	0.5547	0.5842	0.5402	0.8046
T3/T4/T6/Non-effector (4 class)	CQNR	0.4862	0.7827	0.5137	0.6137	0.5517	**0.8234**
	Unbalanced	0.3965	0.6302	0.5308	0.5294	0.5301	0.7193
	Random oversampling	0.4427	0.6729	0.5403	0.5328	0.5673	0.7527
	SMOTE	0.4395	0.6528	0.5583	0.5495	0.5595	0.7527
	Borderline-SMOTE	0.4436	0.7193	0.5245	0.5574	0.5247	0.7895
	C-SMOTE	0.4539	0.7527	0.5139	0.5603	0.5385	0.7627
	Safe-level-SMOTE	0.4293	0.7627	0.5402	0.5782	0.5467	0.8046

Table A.9: Summary of Cohen's (κ) score of the classifiers on the experimentally verified pathogenic effector proteins after 20 fold cross-validation before and after dataset balancing. '+' indicates the classes merged into a single class. '/' indicates that the classes on either side of '/' are treated as a separate class.

Effector Protein set	Method	SVM	MLP	NB	kNN	RF	Consensus
T3/T4/T6 (3 class)	CQNR	0.5932	0.5085	**0.5964**	0.5896	0.5734	0.5932
	Unbalanced	0.3965	0.5308	0.4937	0.5301	0.5217	0.5357
	Random oversampling	0.4427	0.5403	0.5638	0.5673	0.5323	0.5620
	SMOTE	0.4395	0.5583	0.5205	0.5595	0.5493	0.5432
	Borderline-SMOTE	0.4436	0.5245	0.5893	0.5247	0.5577	0.5153
	C-SMOTE	0.4539	0.5139	0.5428	0.5385	0.5247	0.5249
	Safe-level-SMOTE	0.4293	0.5402	0.5638	0.5467	0.5683	0.5329
T3/(T4+T6) (2 class)	CQNR	0.5708	0.5835	0.6258	0.5730	0.6019	**0.6643**
	Unbalanced	0.5250	0.5163	0.5205	0.5250	0.5564	0.5605
	Random oversampling	0.5129	0.5120	0.5395	0.5129	0.5632	0.5795
	SMOTE	0.5353	0.4932	0.5428	0.5353	0.5796	0.5828
	Borderline-SMOTE	0.5403	0.5053	0.5638	0.5403	0.5932	0.5938
	C-SMOTE	0.5587	0.5249	0.6301	0.5587	0.5823	0.6301
	Safe-level-SMOTE	0.5435	0.5129	0.5893	0.5435	0.5238	0.5893
T4/(T3+T6) (2 class)	CQNR	0.5866	0.5786	0.5650	0.5969	0.5917	**0.6154**
	Unbalanced	0.4937	0.5217	0.5016	0.4825	0.4345	0.5425
	Random oversampling	0.5638	0.5284	0.5120	0.5205	0.5538	0.5550
	SMOTE	0.5205	0.5382	0.4932	0.5893	0.5105	0.5924
	Borderline-SMOTE	0.5893	0.5423	0.5053	0.5205	0.5893	0.5734
	C-SMOTE	0.5428	0.5529	0.5249	0.5638	0.5123	0.5665
	Safe-level-SMOTE	0.5638	0.5386	0.5129	0.5428	0.5356	0.5824
T6/(T3+T4) (2 class)	CQNR	0.6975	0.6632	0.6925	0.6717	0.6613	**0.7425**
	Unbalanced	0.5428	0.5157	0.5851	0.5428	0.5539	0.6741
	Random oversampling	0.5924	0.5638	0.5943	0.5924	0.5686	0.6527
	SMOTE	0.6302	0.5428	0.6081	0.6302	0.5542	0.5730
	Borderline-SMOTE	0.6193	0.5205	0.6372	0.6193	0.5724	0.6301
	C-SMOTE	0.5638	0.5638	0.6153	0.5638	0.5932	0.6193
	Safe-level-SMOTE	0.6527	0.5924	0.6005	0.6527	0.5842	0.6301
T3/T4 (2 class)	CQNR	0.5975	0.5766	0.5312	0.5841	0.5783	**0.6286**
	Unbalanced	0.5467	0.5634	0.5308	0.5308	0.5301	0.5539
	Random oversampling	0.5583	0.5539	0.5403	0.5403	0.5673	0.5686
	SMOTE	0.5639	0.5686	0.5583	0.5583	0.5595	0.5542
	Borderline-SMOTE	0.5783	0.5542	0.5245	0.5245	0.5247	0.5724
	C-SMOTE	0.5428	0.5724	0.5139	0.5139	0.5385	0.5932
	Safe-level-SMOTE	0.5547	0.5932	0.5402	0.5402	0.5467	0.5842
T3/T6 (2 class)	CQNR	0.5791	0.5576	0.5433	0.6009	0.6518	**0.6957**
	Unbalanced	0.5308	0.5301	0.5467	0.5467	0.5851	0.5294
	Random oversampling	0.5403	0.5673	0.5583	0.5583	0.5943	0.5728
	SMOTE	0.5583	0.5595	0.5639	0.5639	0.5081	0.5495
	Borderline-SMOTE	0.5245	0.5247	0.5783	0.5783	0.5372	0.5874
	C-SMOTE	0.5139	0.5385	0.5428	0.5428	0.5153	0.5603
	Safe-level-SMOTE	0.5402	0.5467	0.5547	0.5547	0.5005	0.5782
T4/T6 (2 class)	CQNR	0.5950	0.6070	0.6025	0.5752	0.6521	**0.7494**
	Unbalanced	0.5016	0.5129	0.5205	0.5129	0.5632	0.5329
	Random oversampling	0.5172	0.5353	0.5893	0.5353	0.5796	0.5495
	SMOTE	0.5269	0.5403	0.5205	0.5403	0.5932	0.5502
	Borderline-SMOTE	0.5307	0.5587	0.5638	0.5587	0.5823	0.5675
	C-SMOTE	0.5537	0.5435	0.5428	0.5435	0.5238	0.5758
	Safe-level-SMOTE	0.5264	0.5791	0.5205	0.5791	0.6009	0.6918
T3/T4/T6/Other/Non-effector (5 class)	CQNR	0.6268	0.5668	0.5868	0.5756	0.5887	**0.6937**
	Unbalanced	0.5329	0.5139	0.5402	0.5172	0.5402	0.5329
	Random oversampling	0.5402	0.5632	0.5307	0.5632	0.5139	0.5732
	SMOTE	0.5307	0.5172	0.5587	0.5139	0.5307	0.5672
	Borderline-SMOTE	0.5139	0.5402	0.5139	0.5329	0.5329	0.5802
	C-SMOTE	0.5172	0.5329	0.5632	0.5307	0.5632	0.5939
	Safe-level-SMOTE	0.5632	0.5307	0.5172	0.5587	0.5172	0.6225
T3/Non-effectors (2 class)	CQNR	0.5527	0.6361	0.5861	0.5944	0.6083	**0.7913**
	Unbalanced	0.4923	0.5638	0.5583	0.5403	0.5403	0.6838
	Random oversampling	0.4625	0.5205	0.5639	0.5683	0.5583	0.7193
	SMOTE	0.4959	0.5893	0.5783	0.5245	0.5245	0.695
	Borderline-SMOTE	0.5329	0.5428	0.5428	0.5139	0.5139	0.7556
	C-SMOTE	0.5623	0.5638	0.5547	0.5402	0.5402	0.7494
	Safe-level-SMOTE	0.5416	0.6116	0.5421	0.5684	0.5568	0.7361

Table A.9 continues: Summary of Cohen's (κ) score of the classifiers on the experimentally verified pathogenic effector proteins after 20 fold cross-validation before and after dataset balancing. '+' indicates the classes merged into a single class. '/' indicates that the classes on either side of '/' are treated as a separate class.

Effector Protein set	Method	SVM	MLP	NB	kNN	RF	Consensus
T4/Non-effectors (2 class)	CQNR	0.6027	0.6301	0.5794	0.5205	0.6438	**0.6893**
	Unbalanced	0.5583	0.5539	0.5239	0.512	0.5428	0.5924
	Random oversampling	0.5245	0.5686	0.532	0.4932	0.5924	0.6301
	SMOTE	0.5139	0.5542	0.5473	0.5053	0.6301	0.6193
	Borderline-SMOTE	0.5402	0.5724	0.532	0.5249	0.6193	0.5638
	C-SMOTE	0.5568	0.5932	0.5534	0.5129	0.5638	0.6527
	Safe-level-SMOTE	0.6083	0.5842	0.5576	0.5433	0.6527	0.6741
T6/Non-effector (2 class)	CQNR	0.6638	0.7750	0.6416	0.7387	0.8033	**0.8523**
	Unbalanced	0.5238	0.6757	0.5157	0.6838	0.6943	0.6943
	Random oversampling	0.5329	0.6823	0.5923	0.7193	0.7024	0.7324
	SMOTE	0.5495	0.6946	0.5625	0.7494	0.7394	0.7594
	Borderline-SMOTE	0.5502	0.7023	0.5959	0.7556	0.7556	0.7556
	C-SMOTE	0.5675	0.6736	0.6329	0.7694	0.7723	0.7753
	Safe-level-SMOTE	0.5758	0.7020	0.6230	0.7777	0.7446	0.7846
Other/Non-effector (2 class)	CQNR	0.7472	0.8245	0.5138	0.7916	0.7166	**0.8623**
	Unbalanced	0.6757	0.6943	0.5238	0.7193	0.6239	0.7827
	Random oversampling	0.6823	0.7024	0.5329	0.6950	0.6927	0.8025
	SMOTE	0.6946	0.7394	0.5495	0.7556	0.6638	0.8237
	Borderline-SMOTE	0.7023	0.7556	0.5502	0.7494	0.7623	0.8134
	C-SMOTE	0.6736	0.7723	0.5675	0.7361	0.6827	0.7827
	Safe-level-SMOTE	0.7020	0.7446	0.5758	0.7423	0.7025	0.8237
Eff/Non-effector (2 class)	CQNR	0.58	0.5533	0.5145	**0.6990**	0.6116	0.6423
	Unbalanced	0.5403	0.5308	0.5012	0.5239	0.5205	0.5638
	Random oversampling	0.5583	0.5403	0.5120	0.5356	0.5395	0.5205
	SMOTE	0.5245	0.5583	0.4932	0.5406	0.5428	0.5893
	Borderline-SMOTE	0.5139	0.5245	0.5053	0.5912	0.5638	0.5428
	C-SMOTE	0.5402	0.5139	0.5249	0.6234	0.6301	0.5638
	Safe-level-SMOTE	0.5533	0.5402	0.5129	0.6766	0.5893	0.6116
(T3+T4+T6)/Non-effector (2 class)	CQNR	0.5393	0.573	0.5617	0.5842	0.6741	**0.7323**
	Unbalanced	0.4666	0.5294	0.5217	0.5034	0.5851	0.5071
	Random oversampling	0.4738	0.5328	0.53	0.5273	0.5943	0.5283
	SMOTE	0.4823	0.5495	0.5493	0.5139	0.6081	0.5832
	Borderline-SMOTE	0.4902	0.5574	0.5577	0.5467	0.6372	0.6273
	C-SMOTE	0.5022	0.5603	0.5247	0.5305	0.6153	0.6193
	Safe-level-SMOTE	0.5137	0.5782	0.5683	0.5734	0.6005	0.6593
(T3+T4+T6)/Other (2 class)	CQNR	0.7181	0.8318	0.5568	0.7727	0.7636	0.8624
	Unbalanced	0.5943	0.6239	0.4923	0.7193	0.695	0.8025
	Random oversampling	0.6081	0.6927	0.4625	0.7494	0.6838	0.8237
	SMOTE	0.6372	0.7638	0.4959	0.7361	0.7193	0.8134
	Borderline-SMOTE	0.6153	0.8623	0.5329	0.7694	0.7494	0.7827
	C-SMOTE	0.6005	0.7827	0.5623	0.7556	0.7556	0.8237
	Safe-level-SMOTE	0.6804	0.8025	0.5416	0.7638	0.7694	**0.8772**
T3/T4/T6/Other (4 class)	CQNR	0.5473	0.6368	0.5421	0.6052	0.5684	**0.7723**
	Unbalanced	0.5263	0.6634	0.5403	0.5403	0.5294	0.695
	Random oversampling	0.5173	0.6724	0.5583	0.5583	0.5683	0.6838
	SMOTE	0.5212	0.6264	0.5245	0.5245	0.5245	0.7193
	Borderline-SMOTE	0.6556	0.7556	0.5139	0.5139	0.5139	0.7494
	C-SMOTE	0.5018	0.6494	0.5402	0.5402	0.5402	0.7556
	Safe-level-SMOTE	0.4634	0.6361	0.5533	0.5393	0.5684	0.7694
T3/T4/T6/Non-effector (4 class)	CQNR	0.5862	0.6827	0.5537	0.6137	0.5917	**0.8034**
	Unbalanced	0.5375	0.6274	0.5263	0.5375	0.5734	0.6724
	Random oversampling	0.5753	0.6874	0.5437	0.5734	0.5264	0.6365
	SMOTE	0.5385	0.6274	0.5328	0.5934	0.5635	0.6024
	Borderline-SMOTE	0.5334	0.6658	0.5277	0.5024	0.5254	0.6536
	C-SMOTE	0.5629	0.6384	0.5163	0.5947	0.5635	0.6845
	Safe-level-SMOTE	0.5726	0.6294	0.4634	0.6074	0.5524	0.7745

Table A.10: Summary of *MCC* of the classifiers on the experimentally verified pathogenic effector proteins after 20 fold cross-validation before and after dataset balancing. '+' indicates the classes merged into a single class. '/' indicates that the classes on either side of '/' are treated as a separate class.

Effector Protein set	Method	SVM	MLP	NB	kNN	RF	Consensus
T3/T4/T6 (3 class)	CQNR	0.5932	0.5085	0.5964	0.5896	0.5734	0.5432
	Unbalanced	0.3345	0.5152	0.4273	0.5243	0.5184	0.5172
	Random oversampling	0.4134	0.5135	0.5173	0.5085	0.536	0.5273
	SMOTE	0.4264	0.5042	0.5232	0.5146	0.5283	0.4345
	Borderline-SMOTE	0.4376	0.5064	0.5186	0.5163	0.5593	**0.5932**
	C-SMOTE	0.4684	0.5345	0.5306	0.5664	0.5418	0.5832
	Safe-level-SMOTE	0.4934	0.5246	0.5479	0.5517	0.5593	0.5214
T3/(T4+T6) (2 class)	CQNR	0.5708	0.5835	0.6258	0.573	0.6019	**0.6643**
	Unbalanced	0.5182	0.5246	0.5384	0.5264	0.5642	0.5253
	Random oversampling	0.5621	0.5424	0.5283	0.5274	0.5424	0.5255
	SMOTE	0.5421	0.479	0.5736	0.5532	0.5545	0.5563
	Borderline-SMOTE	0.524	0.523	0.5824	0.5232	0.5256	0.5345
	C-SMOTE	0.5453	0.5435	0.6023	0.5532	0.5654	0.6253
	Safe-level-SMOTE	0.5356	0.5345	0.5925	0.5232	0.5245	0.5633
T4/(T3+T6) (2 class)	CQNR	0.4866	0.5786	0.5650	0.5969	0.5317	**0.6387**
	Unbalanced	0.4634	0.5183	0.5412	0.4374	0.4234	0.5345
	Random oversampling	0.5294	0.5194	0.5173	0.5824	0.5425	0.5214
	SMOTE	0.5194	0.5144	0.4627	0.5764	0.5264	0.5634
	Borderline-SMOTE	0.5644	0.5324	0.5523	0.5643	0.5234	0.5245
	C-SMOTE	0.5834	0.5130	0.5334	0.5356	0.5624	0.5562
	Safe-level-SMOTE	0.5183	0.5423	0.5234	0.5356	0.5210	0.5254
T6/(T3+T4) (2 class)	CQNR	0.6975	0.6632	0.6925	0.6717	0.6613	**0.7225**
	Unbalanced	0.5244	0.534	0.5345	0.5535	0.5566	0.6454
	Random oversampling	0.5632	0.5525	0.5534	0.5645	0.5345	0.6345
	SMOTE	0.6123	0.5325	0.6345	0.6245	0.5234	0.5563
	Borderline-SMOTE	0.6214	0.5256	0.6530	0.6234	0.5542	0.6462
	C-SMOTE	0.5524	0.5534	0.6646	0.5534	0.5355	0.6346
	Safe-level-SMOTE	0.6145	0.5632	0.6765	0.6234	0.5645	0.6234
T3/T4 (2 class)	CQNR	0.5975	0.5766	0.5312	0.5841	0.5483	**0.6286**
	Unbalanced	0.5466	0.5483	0.5372	0.5265	0.5467	0.5472
	Random oversampling	0.5365	0.5743	0.5473	0.556	0.5745	0.5527
	SMOTE	0.5357	0.5453	0.5564	0.5628	0.5468	0.5742
	Borderline-SMOTE	0.5653	0.5375	0.5746	0.5621	0.5295	0.5327
	C-SMOTE	0.5735	0.5654	0.5475	0.5251	0.5372	0.5960
	Safe-level-SMOTE	0.5375	0.5968	0.5364	0.5387	0.5062	0.5275
T3/T6 (2 class)	CQNR	0.5791	0.5576	0.5433	0.6009	0.6918	**0.6557**
	Unbalanced	0.5258	0.5107	0.5186	0.5538	0.5375	0.5638
	Random oversampling	0.5338	0.5046	0.5306	0.5386	0.5673	0.5643
	SMOTE	0.5037	0.5153	0.5185	0.5073	0.6361	0.5357
	Borderline-SMOTE	0.5185	0.5185	0.5063	0.5164	0.6537	0.5683
	C-SMOTE	0.5319	0.5385	0.5026	0.5742	0.6274	0.5724
	Safe-level-SMOTE	0.5386	0.5267	0.5174	0.5386	0.6374	0.5942
T4/T6 (2 class)	CQNR	0.595	0.607	0.6025	0.5752	0.6521	**0.7494**
	Unbalanced	0.5364	0.5464	0.5543	0.5042	0.5065	0.5664
	Random oversampling	0.5174	0.5318	0.5753	0.5275	0.5042	0.5597
	SMOTE	0.5065	0.5042	0.5218	0.5218	0.5473	0.5065
	Borderline-SMOTE	0.5275	0.5749	0.5543	0.5597	0.5187	0.5042
	C-SMOTE	0.5463	0.5473	0.5364	0.5473	0.5364	0.5473
	Safe-level-SMOTE	0.5597	0.5836	0.5275	0.5065	0.6172	0.6824
T3/T4/T6/Other/Non-effector (5 class)	CQNR	0.6268	0.5668	0.5368	0.5756	0.5587	**0.6521**
	Unbalanced	0.5065	0.5543	0.5319	0.5174	0.5473	0.5174
	Random oversampling	0.5244	0.5473	0.5065	0.5244	0.5643	0.5643
	SMOTE	0.5473	0.5319	0.5174	0.5473	0.5174	0.5319
	Borderline-SMOTE	0.5174	0.5065	0.5543	0.5319	0.5244	0.5273
	C-SMOTE	0.5319	0.5643	0.5473	0.5643	0.5543	0.5175
	Safe-level-SMOTE	0.5543	0.5174	0.5244	0.5065	0.5319	0.5386
T3/Non-effectors (2 class)	CQNR	0.5527	0.6361	0.5861	0.5944	0.6083	**0.8026**
	Unbalanced	0.4374	0.5065	0.5749	0.5473	0.5749	0.6245
	Random oversampling	0.424	0.5042	0.5319	0.5065	0.5543	0.7374
	SMOTE	0.4264	0.5473	0.5543	0.5174	0.5174	0.6953
	Borderline-SMOTE	0.5543	0.5174	0.5065	0.5319	0.5319	0.7157
	C-SMOTE	0.5473	0.5319	0.5042	0.5543	0.5042	0.7422
	Safe-level-SMOTE	0.5319	0.6245	0.5473	0.5042	0.5473	0.7275

Table A.10 continues: Summary of *MCC* of the classifiers on the experimentally verified pathogenic effector proteins after 20 fold cross-validation before and after dataset balancing. '+' indicates the classes merged into a single class. '/' indicates that the classes on either side of '/' are treated as a separate class.

Effector Protein set	Method	SVM	MLP	NB	kNN	RF	Consensus
T4/Non-effectors (2 class)	CQNR	0.6027	0.6301	0.5794	0.5205	0.6438	**0.7038**
	Unbalanced	0.5127	0.5319	0.5042	0.5543	0.5543	0.5319
	Random oversampling	0.5163	0.5042	0.5473	0.4524	0.5473	0.6083
	SMOTE	0.5335	0.5473	0.5319	0.5042	0.6361	0.5861
	Borderline-SMOTE	0.534	0.5543	0.5543	0.5065	0.6083	0.5473
	C-SMOTE	0.5135	0.5042	0.5065	0.5319	0.5319	0.6361
	Safe-level-SMOTE	0.6083	0.5065	0.5042	0.5473	0.5861	0.6124
T6/Non-effector (2 class)	CQNR	0.6638	0.775	0.6416	0.7777	**0.8233**	0.8123
	Unbalanced	0.5135	0.6124	0.5263	0.6124	0.6124	0.6124
	Random oversampling	0.5263	0.6083	0.5135	0.7275	0.7422	0.7157
	SMOTE	0.5042	0.6361	0.5042	0.7422	0.7264	0.7275
	Borderline-SMOTE	0.5543	0.7422	0.5543	0.7264	0.7157	0.7264
	C-SMOTE	0.5319	0.6027	0.6124	0.7157	0.7275	0.7422
	Safe-level-SMOTE	0.534	0.7157	0.6083	0.7422	0.7024	0.7246
Other/Non-effector (2 class)	CQNR	0.7472	0.8203	0.5138	0.7916	0.8166	0.8523
	Unbalanced	0.6638	0.6416	0.5543	0.7157	0.6638	0.7422
	Random oversampling	0.6124	0.7157	0.5263	0.6638	0.6361	0.8385
	SMOTE	0.6361	0.7264	0.5135	0.7422	0.7422	0.8183
	Borderline-SMOTE	0.7264	0.7264	0.5042	0.7422	0.8385	0.8274
	C-SMOTE	0.6027	0.7422	0.5263	0.7264	0.7264	0.7264
	Safe-level-SMOTE	0.7422	0.7422	0.5135	0.7157	0.8183	**0.8632**
Eff/Non-effector (2 class)	CQNR	0.58	0.5533	0.5145	0.6910	0.6116	0.6423
	Unbalanced	0.5263	0.5319	0.5319	0.5263	0.5543	0.5135
	Random oversampling	0.5135	0.5138	0.5543	0.5135	0.5065	0.5138
	SMOTE	0.5319	0.5543	0.5263	0.5065	0.5138	0.5263
	Borderline-SMOTE	0.5543	0.5065	0.5135	0.5543	0.5135	0.5065
	C-SMOTE	0.5065	0.5263	0.5065	0.6374	0.6012	0.5543
	Safe-level-SMOTE	0.5138	0.5135	0.5138	**0.6927**	0.5263	0.6273
(T3+T4+T6)/Non-effector (2 class)	CQNR	0.5393	0.573	0.5617	0.5842	0.6741	**0.7323**
	Unbalanced	0.4036	0.5263	0.5543	0.5135	0.5065	0.5543
	Random oversampling	0.4624	0.5462	0.5135	0.5263	0.5543	0.5065
	SMOTE	0.424	0.5065	0.5042	0.5543	0.6264	0.5263
	Borderline-SMOTE	0.449	0.5543	0.5263	0.5065	0.6374	0.6927
	C-SMOTE	0.4374	0.5135	0.5263	0.5042	0.6427	0.6273
	Safe-level-SMOTE	0.5543	0.5042	0.5065	0.5462	0.6273	0.6374
(T3+T4+T6)/Other (2 class)	CQNR	0.7181	0.8318	0.5568	0.7727	0.7636	0.8624
	Unbalanced	0.5163	0.6264	0.4374	0.7323	0.6374	0.8013
	Random oversampling	0.6264	0.6374	0.4240	0.7527	0.6264	0.8276
	SMOTE	0.6374	0.7323	0.4352	0.7157	0.7422	0.8519
	Borderline-SMOTE	0.6741	0.8013	0.5320	0.7163	0.7275	0.7323
	C-SMOTE	0.6427	0.7157	0.5163	0.7422	0.7157	**0.8713**
	Safe-level-SMOTE	0.6264	0.8276	0.5206	0.7275	0.7323	0.8442
T3/T4/T6/Other (4 class)	CQNR	0.5473	0.7368	0.5421	0.6052	0.5684	**0.7723**
	Unbalanced	0.5163	0.6432	0.5140	0.5532	0.5253	0.6532
	Random oversampling	0.5200	0.7130	0.5742	0.5136	0.5134	0.6574
	SMOTE	0.5320	0.6145	0.5315	0.5521	0.5134	0.7246
	Borderline-SMOTE	0.5135	0.7216	0.5162	0.5148	0.5245	0.7421
	C-SMOTE	0.5210	0.7174	0.5320	0.5258	0.5421	0.7245
	Safe-level-SMOTE	0.4145	0.7216	0.5162	0.5326	0.5234	0.7346
T3/T4/T6/Non-effector (4 class)	CQNR	0.5862	0.7827	0.5137	0.6137	0.5517	**0.7934**
	Unbalanced	0.5364	0.7264	0.5147	0.5235	0.5231	0.6247
	Random oversampling	0.5230	0.7213	0.5416	0.5636	0.5532	0.6853
	SMOTE	0.5120	0.7424	0.5462	0.5525	0.5124	0.6632
	Borderline-SMOTE	0.5235	0.7296	0.5467	0.5247	0.5406	0.6753
	C-SMOTE	0.5164	0.7483	0.5124	0.5742	0.5101	0.6753
	Safe-level-SMOTE	0.5432	0.7147	0.4257	0.6012	0.5320	0.7264

Table A.11: Performance comparison of T3 effector protein predictors on an independent set of proteins.

Effector protein ID (PDB)	EPP3D	DeepT3	Bastion3	Wang *et al.*
5T09	Effector	Effector	Effector	Effector
2WUN	Effector	Effector	Effector	Effector
2QMZ	Effector	Non-effector	Effector	Non-effector
4QMK	Effector	Effector	Effector	Non-effector
1S21	Effector	Effector	Effector	Effector
3I0U	Non-effector	Effector	Effector	Non-effector
6CJD	Effector	Non-effector	Effector	Effector
1NH1	Effector	Effector	Effector	Non-effector
2NUD	Non-effector	Effector	Effector	Non-effector
4FC9	Effector	Non-effector	Effector	Effector
4FCG	Non-effector	Non-effector	Effector	Non-effector
6AE1	Non-effector	Effector	Non-effector	Effector
6AE2	Effector	Effector	Non-effector	Effector
3CKD	Effector	Effector	Effector	Effector
6IQW	Effector	Effector	Non-effector	Effector
2KQ5	Non-effector	Non-effector	Effector	Non-effector
3EE1	Effector	Non-effector	Effector	Non-effector
4P5F	Effector	Non-effector	Effector	Effector
6HQZ	Effector	Effector	Effector	Non-effector
1R5E	Effector	Non-effector	Effector	Effector

Table A.12: Performance comparison of T4 effector protein predictors on an independent set of proteins.

Effector protein ID (PDB)	EPP3D	Zou *et al.*	Xiong *et al.*	Bastion4	Burstein *et al.*
2VY3	Effector	Effector	Effector	Undefined	Effector
2VZA	Effector	Non-effector	Effector	Undefined	Non-effector
2JK8	Effector	Non-effector	Effector	Undefined	Effector
6H94	Effector	Effector	Effector	Undefined	Non-effector
6HPI	Effector	Non-effector	Non-effector	Undefined	Effector
3L0I	Effector	Effector	Effector	Undefined	Non-effector
5CZY	Effector	Non-effector	Non-effector	Undefined	Effector
4YK3	Effector	Effector	Effector	Undefined	Effector
1S21	Effector	Effector	Non-effector	Undefined	Effector
2NUD	Effector	Non-effector	Non-effector	Undefined	Non-effector
3L0M	Non-effector	Effector	Non-effector	Undefined	Non-effector
4BED	Effector	Effector	Effector	Undefined	Effector
4BER	Effector	Effector	Effector	Undefined	Non-effector
4BES	Effector	Effector	Effector	Undefined	Effector
5X1E	Effector	Non-effector	Effector	Undefined	Non-effector
5X1H	Effector	Non-effector	Effector	Undefined	Effector
5X1U	Non-effector	Non-effector	Non-effector	Undefined	Non-effector
5X42	Effector	Effector	Effector	Undefined	Effector
5X90	Effector	Effector	Effector	Undefined	Effector
4FGI	Effector	Effector	Non-effector	Undefined	Non-effector

Table A.13: Performance comparison of T6 effector protein predictors on an independent set of proteins.

Effector protein ID (PDB)	EPP3D	Bastion6	PyPredT6
3V4H	Effector	Undefined	Effector
3VPI	Effector	Undefined	Effector
4EOB	Effector	Undefined	Effector
4HFL	Effector	Undefined	Effector
4FOV	Effector	Undefined	Effector
4FOW	Effector	Undefined	Non-effector
6IJE	Non-effector	Undefined	Effector
3VPJ	Effector	Undefined	Effector
3WA5	Effector	Undefined	Non-effector
4BI8	Effector	Undefined	Non-effector
4F4M	Effector	Undefined	Effector
4HFF	Effector	Undefined	Effector
4HFK	Effector	Undefined	Effector
4HZ9	Effector	Undefined	Effector
4HZB	Non-effector	Undefined	Non-effector
4KT3	Non-effector	Undefined	Non-effector
6IJF	Effector	Undefined	Effector
4NS0	Effector	Undefined	Effector
6H3L	Effector	Undefined	Non-effector
6H3N	Effector	Undefined	Non-effector

A.3 Chapter 5

A.3.1 Amino acid values for physicochemical properties

With respect to the physicochemical properties numbered 18 to 38, which have contributed to 55 features, described in Table 5.1 of Chapter 5, the feature values corresponding to amino acids with respect to these features have been furnished in Tables A.14 and A.15.

A.3.2 Parameters of the classifiers considered

For SVM, we have considered the kernel function to be the radial basis function (RBF), and two parameters have been considered, which are the regularization parameter C and the kernel width parameter γ, optimized by using a grid search approach. For RF, the two parameters, $mtry$ (the number of variables randomly selected as candidates at each node) and $ntree$ (the number of trees to grow) have been considered; the value of $ntree$ was from 500 to 3000 with a step length of 500, and the value of $mtry$ was from 2 to 50 with a step length of 2. For k-NN, we chose Euclidean distance as distance function and set the number of neighbors (K) in the set 3, 5, 7, 9, 11, 13, 15, 17, 19, and 21. The k value of 11, having the highest prediction performance, was considered. For decision tree classifier (DT), the maximum depth of the tree max_depth was set to 5, and the minimum number of samples required to split an internal node $min\_samples\_split$ was set to 2.

Table A.14: Values of physicochemical properties for amino acids A, L, R, K, N, M, D, C, F, and P.

Physicochemical properties	A	L	R	K	N	M	D	C	F	P
Hydrophobicity factor	0.75	2.4	0.75	1.5	0.69	1.3	0	1	2.65	2.6
Residue volume	52.6	102	109.1	105.1	75.7	97.7	68.4	68.3	113.9	73.6
Transfer free energy to surface	-0.2	-2.46	-0.12	-0.35	0.08	-1.47	-0.2	-0.45	-2.33	-0.98
Apparent partial specific volume	0.691	0.842	0.728	0.767	0.596	0.709	0.558	0.624	0.756	0.73
Polarizability parameter	0.52	0.98	0.68	0.68	0.76	0.78	0.76	0.62	0.7	0.36
Average volume of buried residue	91.5	167.9	202	171.3	135.2	170.8	124.5	117.7	203.4	129.3
Residue accessible surface area in tripeptide	115	170	225	200	160	185	150	135	210	145
Solvation free energy	0.67	1.9	-2.1	-0.57	-0.6	2.4	-1.2	0.38	2.3	1.2
Molecular weight	89.09	131.17	174.2	146.19	132.12	149.21	133.1	121.15	165.19	115.13
Melting point	297	337	238	224	236	283	270	178	284	222
Percentage of buried residues	51	60	5	3	22	52	19	74	58	25
Percentage of exposed residues	15	16	67	85	49	20	50	5	10	45
Signal sequence helical potential	1.18	3.23	0.2	0.06	0.23	2.67	0.05	1.89	1.96	0.76
Membrane-buried preference parameters	1.56	2.93	0.45	0.15	0.27	2.96	0.14	1.23	2.03	0.76
Average flexibility indices	0.357	0.365	0.529	0.466	0.463	0.295	0.511	0.346	0.314	0.509
Polarizability parameter	0.046	0.186	0.291	0.219	0.134	0.221	0.105	0.128	0.29	0.131
Free energy of solution in water, kcal/mole	-0.368	1.07	-1.03	0	0	0.656	2.06	4.53	1.06	-2.24
Relative mutability	100	40	65	56	134	94	106	20	41	56
Atom-based hydrophobic moment	0	1	10	5.7	1.3	1.9	1.9	0.17	1.1	0.18
Normalized van der Waals volume	1	4	6.13	4.77	2.95	4.43	2.78	2.43	5.89	2.72
Localized electrical effect	-0.01	-0.01	0.04	0	0.06	0.04	0.15	0.12	0.03	0
Partition coefficient	0.28	1	0.1	0.09	0.25	0.74	0.21	0.28	2.18	0.39
Hydration number	1	0.8	2.3	5.3	2.2	0.7	6.5	0.1	1.4	0.9
Heat capacity	29.22	48.03	26.37	57.1	38.3	69.32	37.09	50.7	48.52	36.13
Absolute entropy	30.88	50.62	68.43	63.21	41.7	55.32	40.66	53.83	51.06	39.21
Entropy of formation	154.33	232.3	341.01	300.46	207.9	202.65	194.91	219.79	204.74	179.93
Refractivity	4.34	18.78	26.66	21.29	13.28	21.64	12	35.77	29.4	10.93
Retention coefficient in HPLC, pH7.4	0.5	8.8	0.8	0.1	0.8	4.8	-8.2	-6.8	13.2	6.1
Retention coefficient in HPLC, pH2.1	-0.1	10	-4.5	-3.2	-1.6	7.1	-2.8	-2.2	13.9	8
Principal component I	0.239	0.281	0.211	0.228	0.249	0.253	0.171	0.22	0.234	0.165
Principal component II	0.33	0.129	-0.176	-0.075	-0.233	-0.092	-0.371	0.074	-0.011	0.37
Principal component III	-0.11	-0.008	0.079	0.049	-0.136	-0.041	-0.285	-0.184	0.438	-0.016
Principal component IV	-0.062	-0.264	-0.167	-0.371	0.166	0.077	-0.079	0.38	0.074	-0.036
Molecular descriptor d1	2	5	8	6	5	5	5	3	8	4
Molecular descriptor d2	1	4	7	5	4	4	4	2	8	4
Molecular descriptor d3	2	8	12	10	8	8	8	4	14	8
Molecular descriptor d4	1	4	6	4	4	4	4	2	6	4
Molecular descriptor d5	1	5	8.12	7	5	5.4	5.17	2.33	7	4
Molecular descriptor d6	1	3	6	5	3	3	3	1	6	4
Molecular descriptor d7	1	6	12	9	6	7	6	3	11	4
Molecular descriptor d8	1	1.6	1.5	1.667	1.6	1.6	1.6	1.333	1.75	2
Molecular descriptor d9	2	11.029	12.499	10.363	11.539	9.49	11.539	6.243	14.851	12
Molecular descriptor d10	0	4.729	-4.307	-3.151	-4.178	-2.812	-4.178	-2.243	-4.801	-4
Molecular descriptor d11	1	3.2	3.5	3	3.2	2.8	3.2	2	4.25	4
Molecular descriptor d12	2	1.052	-2.59	-0.536	0.528	0.678	0.528	2	-1.672	4
Molecular descriptor d13	6	12	19	12	12	18	12	6	18	12
Molecular descriptor d14	6	15.6	31.444	24.5	16.5	27.2	16.4	16.67	23.25	12
Molecular descriptor d15	6	12	20	18	14	18	12	12	18	12
Molecular descriptor d16	6	18	38	31	20	34	20	22	24	12
Molecular descriptor d17	12	30	45	37	33.007	40	34	28	48	24
Molecular descriptor d18	6	6	5	6.17	6.6	8	6.8	9.33	6	6
Molecular descriptor d19	12	25.021	23.343	22.739	27.708	31.344	28.634	28	26.993	24
Molecular descriptor d20	0	0	0	-0.179	0	0	0	0	0	0
Molecular descriptor d21	6	9.6	10.667	10.167	10	13.6	10.4	11.333	12	12
Molecular descriptor d22	0	3.113	4.2	1.372	3	2.656	2.969	6	2.026	12

Table A.15: Values of physicochemical properties for amino acids Q, S, E, T, G, W, H, Y, I and V.

Physicochemical properties	Q	S	E	T	G	W	H	Y	I	V
Hydrophobicity factor	0.59	0	0	0.45	0	3	0	2.85	2.95	1.7
Residue volume	89.7	54.9	84.7	71.2	36.3	135.4	91.9	116.2	102	85.1
Transfer free energy to surface	0.16	-0.39	-0.3	-0.52	0	-2.01	-0.12	-2.24	-2.26	-1.56
Apparent partial specific volume	0.649	0.594	0.632	0.632	0.592	0.743	0.646	0.743	0.809	0.777
Steric parameter	0.68	0.53	0.68	0.5	0	0.7	0.7	0.7	1.02	0.76
Average volume of buried residue	161.1	99.1	155.1	122.1	66.4	237.6	167.3	203.6	203.6	141.7
Residue accessible surface area in tripeptide	180	115	190	140	75	255	195	230	175	155
Solvation free energy	-0.22	0.01	-0.76	0.52	0	2.6	0.64	1.6	1.9	1.5
Molecular weight	146.15	105.09	147.13	119.12	75.07	204.24	155.16	181.19	131.17	117.15
Melting point	185	228	249	253	290	282	277	344	284	293
Percentage of buried residues	16	35	16	30	52	49	34	24	66	64
Percentage of exposed residues	56	32	55	32	10	17	34	41	13	14
Signal sequence helical potential	0.72	0.97	0.11	0.84	0.49	0.77	0.31	0.39	1.45	1.08
Membrane-buried preference parameters	0.51	0.81	0.23	0.91	0.62	1.08	0.29	0.68	1.67	1.14
Average flexibility indices	0.493	0.507	0.497	0.444	0.544	0.305	0.323	0.42	0.462	0.386
Polarizability parameter	0.18	0.062	0.151	0.108	0	0.409	0.23	0.298	0.186	0.14
Free energy of solution in water, kcal/mole	0.731	-0.524	1.77	0	-0.525	1.6	0	4.91	0.791	0.401
Relative mutability	93	120	102	97	49	18	66	41	96	74
Atom-based hydrophobic moment	1.9	0.73	3	1.5	0	1.6	0.99	1.8	1.2	0.48
Normalized van der Waals volume	3.95	1.6	3.78	2.6	0	8.08	4.66	6.47	4	3
Localized electrical effect	0.05	0.11	0.07	0.04	0	0	0.08	0.03	-0.01	0.01
Partition coefficient	0.35	0.12	0.33	0.21	0.17	5.7	0.21	1.26	0.82	0.6
Hydration number	2.1	1.7	6.2	1.5	1.1	1.9	2.8	2.1	0.8	0.9
Heat capacity	44.02	32.4	41.84	35.2	23.71	56.92	59.64	51.73	45	40.35
Absolute entropy	46.62	35.65	44.98	36.5	24.74	60	65.99	51.15	49.71	42.75
Entropy of formation	235.51	174.06	223.16	205.8	127.9	237.01	242.54	229.15	233.21	207.6
Refractivity	17.56	6.35	17.26	11.01	0	42.53	21.81	31.53	19.06	13.92
Retention coefficient in HPLC, pH7.4	-4.8	1.2	-16.9	2.7	0	14.9	-3.5	6.1	13.9	2.7
Retention coefficient in HPLC, pH2.1	-2.5	-3.7	-7.5	1.5	-0.5	18.1	0.8	8.2	11.8	3.3
Principal component I	0.26	0.236	0.187	0.213	0.16	0.183	0.205	0.193	0.273	0.255
Principal component II	-0.254	0.022	-0.409	0.136	0.37	-0.011	-0.078	-0.138	0.149	0.245
Principal component III	-0.067	-0.153	-0.246	-0.208	-0.073	0.493	0.32	0.381	0.001	-0.155
Principal component IV	-0.025	0.47	-0.184	0.348	-0.017	0.05	0.056	0.22	-0.309	-0.212
Molecular descriptor d1	6	3	6	4	1	11	7	9	5	4
Molecular descriptor d2	5	2	5	3	0	12	6	9	4	3
Molecular descriptor d3	10	4	10	6	0	24	14	18	8	6
Molecular descriptor d4	4	2	5	3	1	8	6	7	4	3
Molecular descriptor d5	5.86	1.67	6	3.25	0	11.1	6.71	8.88	3.25	3.25
Molecular descriptor d6	4	2	4	1	0	9	6	6	3	1
Molecular descriptor d7	8	3	8	4	0	14	9	13	6	4
Molecular descriptor d8	1.667	1.333	1.667	1.5	0	2.182	2	2	1.6	1.5
Molecular descriptor d9	12.207	5	11.53	9.928	0	13.511	12.876	12.868	10.851	9.928
Molecular descriptor d10	-4.255	1	-3.425	-3.928	0	-6.324	-3.721	-4.793	-6.085	-3.928
Molecular descriptor d11	3.333	2	3.333	3	0	4	4.286	4.333	1.8	3
Molecular descriptor d12	-1.043	2	-0.538	3	0	-2.576	-1.185	-2.054	-1.517	3
Molecular descriptor d13	12	6	12	6	1	24	15	18	12	6
Molecular descriptor d14	21.167	13.33	21	12.4	3.5	27.5	23.1	27.78	15.6	10.5
Molecular descriptor d15	15	8	14	8	1	18	18	20	12	6
Molecular descriptor d16	24	20	26	14	6	36	31	38	18	12
Molecular descriptor d17	39	22	40	27	7	68	47	56	30	24.007
Molecular descriptor d18	6.5	7.33	6.67	5.4	3.5	5.667	4.7	6.22	6	6
Molecular descriptor d19	27.831	20	28.731	23.819	7	29.778	24.243	28.252	24.841	24
Molecular descriptor d20	0	0	0	-4.227	0	0.211	-1.734	-0.96	-1.641	0
Molecular descriptor d21	10.5	8.667	10.667	9	3.5	12.75	10.4	12.222	9.6	9
Molecular descriptor d22	1.849	6	1.822	6	0	2.044	1.605	1.599	3.373	6

A.4 Chapter 6

A.4.1 Database Selection

We have considered KEGG as our primary database, instead of PubChem. Pubchem uses 4 digits to number compounds, which means they can support a maximum of 9999 compounds. KEGG, on the other hand, uses 5 digits to uniquely identify compounds, which means the maximum number of compounds KEGG can support is 99999, which is ten times more than that in Pubchem.

A.4.2 Sample pathway prediction

The formation of six pathways, *viz.*, alpha linoleic acid metabolism, linoleic acid metabolism, glycolysis pathway, TCA cycle, alanine aspartite and glutamate metabolism, and valine, leucine and isoleucine biosynthesis, have been depicted in Figures A.1 to A.6 below.

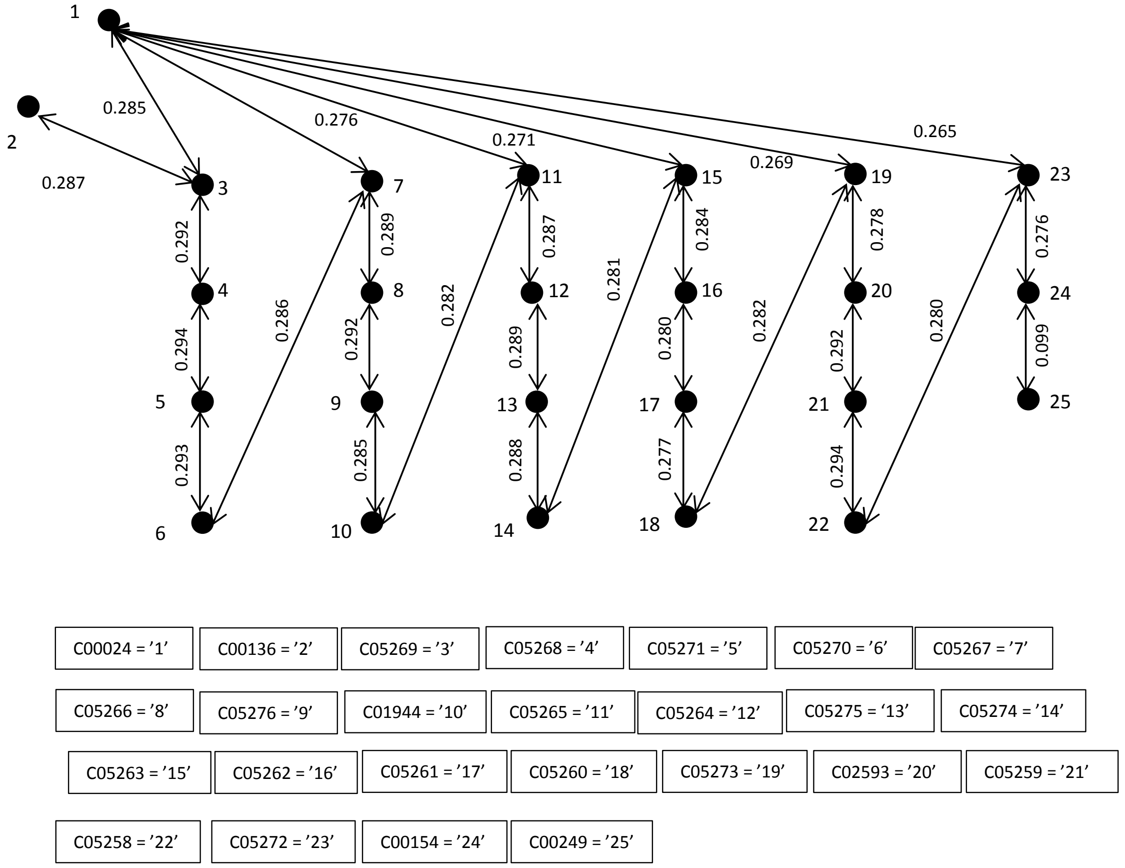
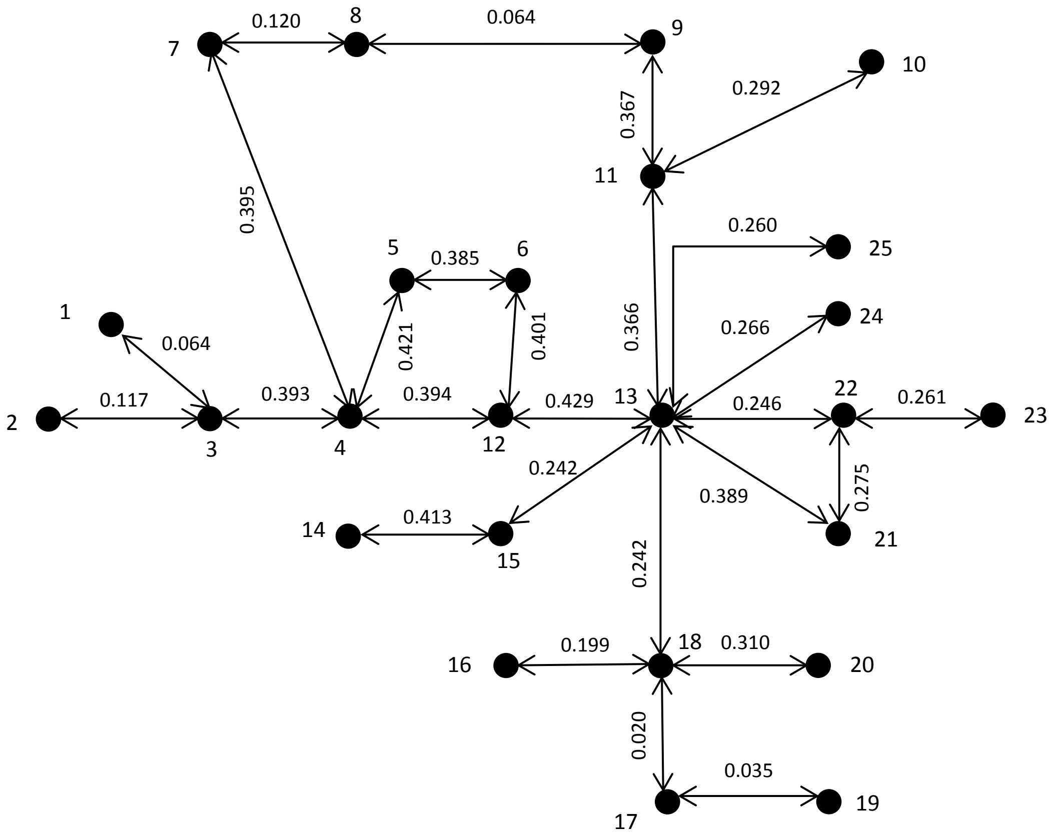


Figure A.1: The full predicted pathway along with the transformation score of the alpha linoleic acid pathway.



C00065 = ’1’ C00154 = ’2’ C02934 = ’3’ C00836 = ’4’ C12144 = ’5’ C12145 = ’6’ C01120 = ’7’

C00346 = ’8’ C06124 = ’9’ C01747 = ’10’ C00319 = ’11’ C12126 = ’12’ C00195 = ’13’ C01290 = ’14’

C01190 = ’15’ C06128 = ’16’ C06126 = ’17’ C02686 = ’18’ C06127 = ’19’ C06125 = ’20’ C02960=’21’

C00550 = ’22’ C03640 = ’23’ C05681 = ’24’ C06062 = ’25’

Figure A.2: The full predicted pathway along with the transformation score of the alpha linoleic acid metabolism pathway.

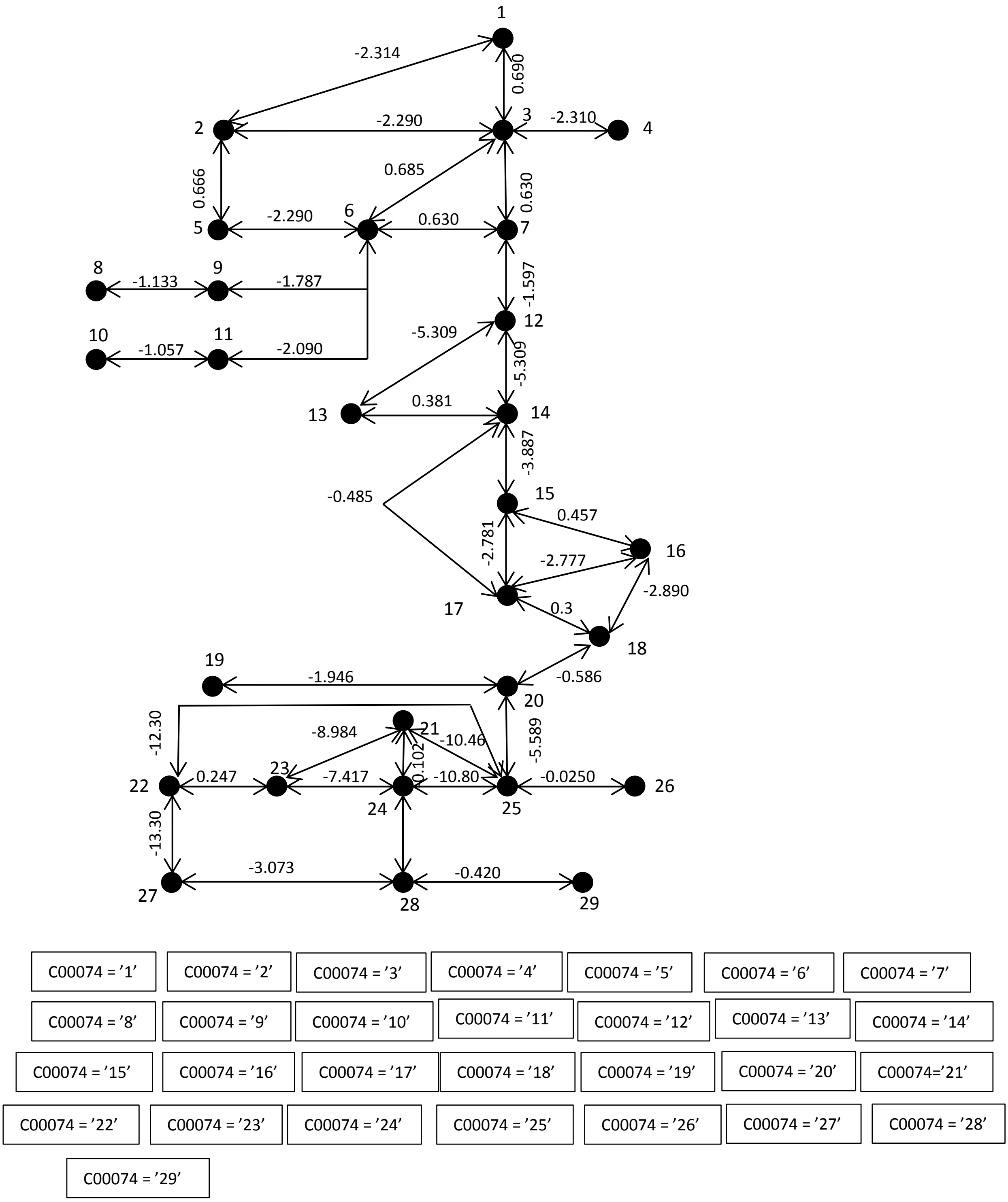
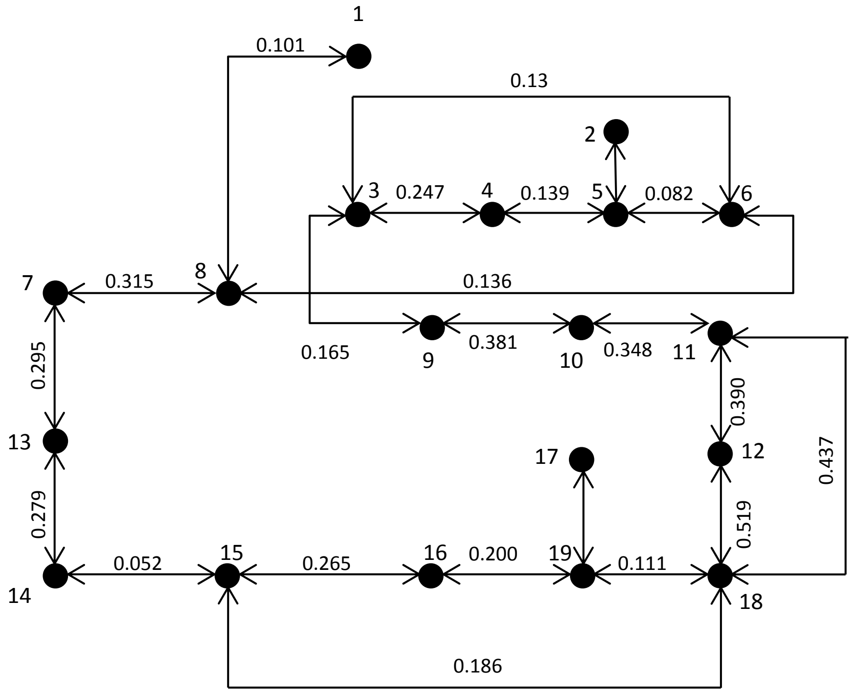


Figure A.3: The full predicted pathway along with the transformation score of the glycolysis pathway.



C00074 = '1' C00068 = '2' C00024 = '3' C16255 = '4' C05125 = '5' C00022 = '6' C00149 = '7'

C00036 = '8' C00158 = '9' C00417 = '10' C00311 = '11' C05379 = '12' C00122 = '13'

C00042 = '14' C00091 = '15' C16254 = '16' C00068 = '17' C00026 = '18' C05381 = '19'

Figure A.4: The full predicted pathway along with the transformation score of the TCA cycle.

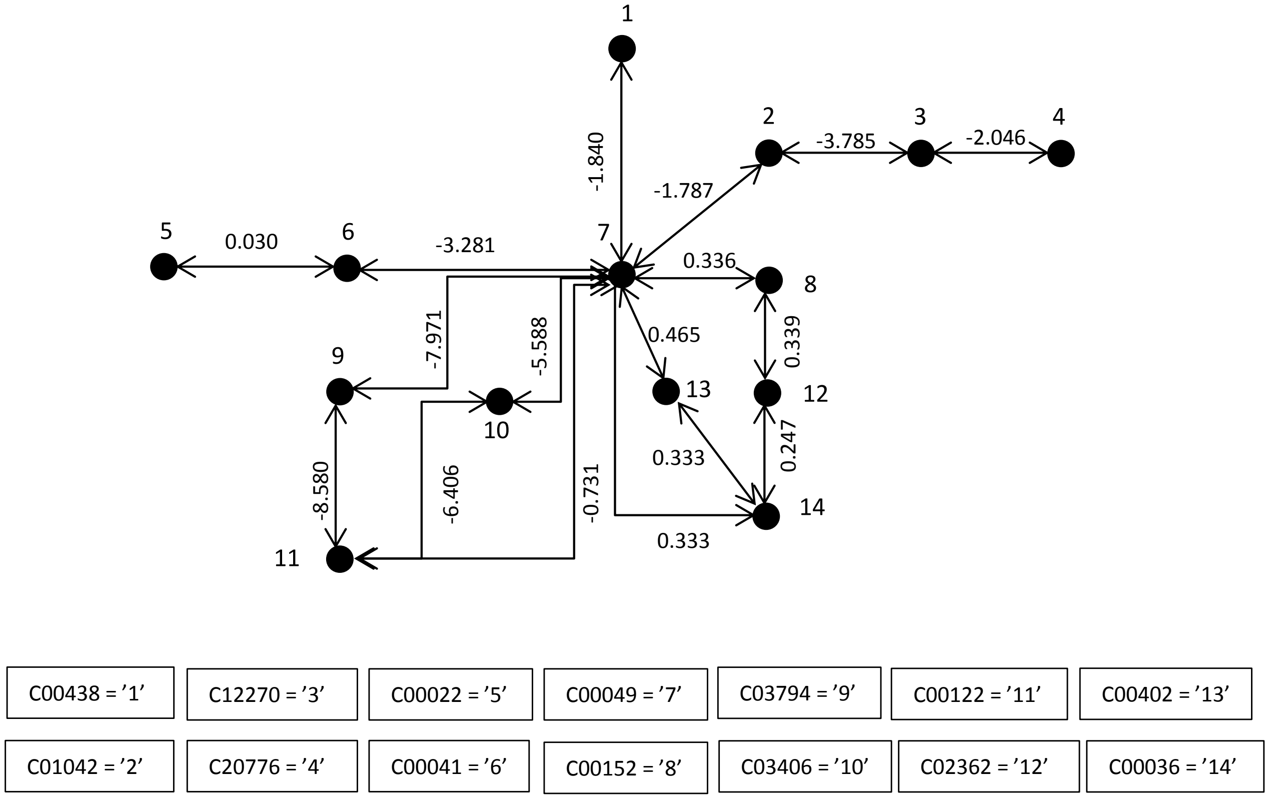


Figure A.5: The full predicted pathway along with the transformation score of the alanine, aspartite and glumate metabolism.

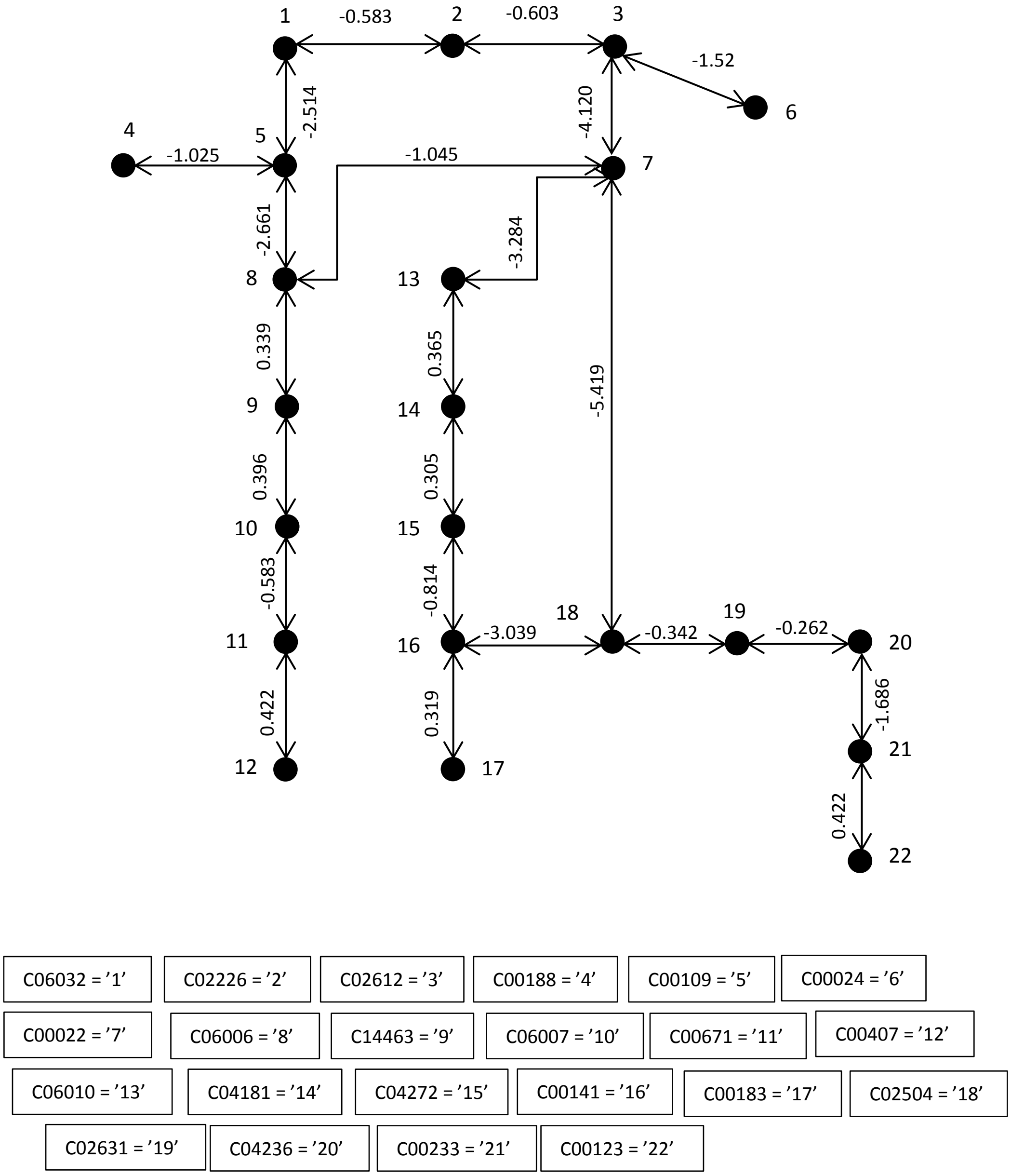


Figure A.6: The full predicted pathway along with the transformation score of the valine, leucine and isoleucine biosynthesis.

A.5 Chapter 7

A.5.1 Technical details of BNRA

BNRA takes the pathways from KEGG in the form of KGML files as input. The KEGG pathway map is a molecular interaction/reaction network diagram represented in terms of the KEGG Orthology (KO) groups, so that experimental evidence in specific organisms can be generalized to other organisms through genomic information. Each pathway map is identified by the combination of 2-4 letter code and 5 digit number (KEGG ID). KGML is an exchange format of KEGG pathway maps[1].

A.5.2 Analysis of 221 pathways from KEGG

We have executed BNRA on 221 pathways which include signal transduction pathways, signaling molecules, and interactions, transportation and metabolism, cell growth and death, cellular community (eukaryotes and prokaryotes) and human diseases. A summary of the results has been given in Table A.16. A summary of drop in stability for each of the 26 groups of the pathways has been given in Table A.17.

[1] https://www.genome.jp/kegg/kegg3a.html

Table A.16: Summary of the results of pathway analysis

KEGG ID	Name	Number of nodes	Number of edges	Components	Max Connectivity	Number of nodes with maximum connectivity	$Rscore$	$PRscore$
ko03015	mRNA surveillance pathway	25	21	8	5	1	0.9690	0.8440
ko04141	Protein processing in endoplasmic reticulum	30	29	5	5	1	0.4763	0.2681
ko04130	SNARE interactions in vesicular transport	14	7	7	1	14	0.7500	0.4464
ko04122	Sulfur relay system	26	20	6	3	1	0.5351	0.2005
ko03440	Homologous recombination	16	16	1	6	1	0.3750	0.0625
ko03460	Fanconi anemia pathway	24	22	2	5	1	0.3796	0.0626
ko02020	Two-component system	478	388	100	16	1	0.4322	0.2350
hsa04014	Ras signaling pathway	93	110	3	18	1	0.0002	6.36E-06
hsa04015	Rap1 signaling pathway	81	90	4	20	1	0.3749	0.1563
hsa04010	MAPK signaling pathway	121	162	5	15	2	0.7140	0.2756
dme04013	MAPK signaling pathway - fly	89	113	5	7	2	0.2953	0.0377
ath04016	MAPK signaling pathway - plant	67	60	11	5	1	0.3200	0.1243
sce04011	MAPK signaling pathway - yeast	107	124	4	9	2	0.0007	9.72E-07
hsa04012	ErbB signaling pathway	56	83	3	13	2	0.4583	0.0645
hsa04310	Wnt signaling pathway	62	71	5	13	1	0.6372	0.3250
hsa04330	Notch signaling pathway	17	16	1	9	1	0.0024	7.63E-06
hsa04340	Hedgehog signaling pathway	46	44	8	7	1	0.6372	0.3212
dme04341	Hedgehog signaling pathway - fly	39	43	7	9	1	0.5513	0.3678
hsa04390	Hippo signaling pathway	76	61	18	7	2	0.6547	0.4128
dme04391	Hippo signaling pathway - fly	52	59	5	13	1	0.5750	0.2000
hsa04392	Hippo signaling pathway - multiple species	37	26	11	4	1	0.6257	0.4112
hsa04370	VEGF signaling pathway	27	25	4	9	1	0.2522	0.0508
hsa04371	Apelin signaling pathway	69	75	2	6	2	0.3416	0.0313
hsa04630	Jak-STAT signaling pathway	33	33	2	17	1	0.2189	0.0156
hsa04064	NF-kappa B signaling pathway	94	84	13	18	1	0.5285	0.3172
hsa04668	TNF signaling pathway	56	53	6	4	5	0.4380	0.2292
hsa04066	HIF-1 signaling pathway	69	67	4	29	1	0.3615	0.0938
hsa04068	FoxO signaling pathway	75	78	3	30	1	0.0832	0.0104
hsa04020	Calcium signaling pathway	22	16	6	7	1	0.5312	0.2083
hsa04072	Phospholipase D signaling pathway	61	71	2	11	1	8.67E-17	2.17E-19
hsa04071	Sphingolipid signaling pathway	63	72	4	8	1	0.2754	0.0314
hsa04024	cAMP signaling pathway	103	114	2	24	1	0.0164	0.0000
hsa04022	cGMP-PKG signaling pathway	72	73	4	18	1	0.4276	0.1094
hsa04151	PI3K-Akt signaling pathway	83	84	4	21	1	0.5097	0.2813
hsa04152	AMPK signaling pathway	67	61	7	22	1	0.6185	0.4212
hsa04150	mTOR signaling pathway	70	82	6	9	2	0.6937	0.2501
hsa04080	Neuroactive ligand-receptor interaction	170	88	82	3	3	0.7362	0.4817
hsa04060	Cytokine-cytokine receptor interaction	260	251	70	9	5	0.7396	0.3541
hsa04512	ECM-receptor interaction	91	94	27	7	1	0.5384	0.2746
hsa04514	Cell adhesion molecules	213	138	85	4	1	0.8560	0.5932

Table A.16 continues: Summary of the results of pathway analysis

KEGG ID	Name	Number of nodes	Number of edges	Components	Max Connectivity	Number of nodes with maximum connectivity	$Rscore$	$PRscore$
hsa04144	Endocytosis	36	24	12	4	1	0.8958	0.7710
hsa04140	Autophagy - animal	87	110	3	11	2	0.7636	0.2500
sce04138	Autophagy - yeast	55	60	5	8	1	0.5832	0.2641
ath04136	Autophagy - other	24	21	6	4	1	0.6464	0.3971
hsa04137	Mitophagy - animal	44	51	2	11	1	2.91E-09	7.28E-10
hsa04110	Cell cycle	72	79	7	9	1	0.7318	0.3571
sce04111	Cell cycle - yeast	83	88	8	6	1	0.4257	0.2070
ccr04112	Cell cycle - Caulobacter	25	32	1	16	1	0.0002	0.0000
sce04113	Meiosis - yeast	72	84	2	10	2	0.2375	0.0313
xla04114	Oocyte meiosis	57	53	6	9	1	0.1849	0.0781
hsa04210	Apoptosis	99	134	2	11	1	0.0244	6.10E-05
dme04214	Apoptosis - fly	62	56	9	6	3	0.3438	0.1147
hsa04216	Ferroptosis	36	28	12	6	1	0.8164	0.6231
hsa04217	Necroptosis	72	76	6	15	1	0.6428	0.1797
hsa04115	p53 signaling pathway	59	63	2	45	1	0.0307	0.0078
hsa04218	Cellular senescence	101	104	16	8	1	0.8725	0.7520
hsa04510	Focal adhesion	61	92	1	11	1	0.0163	0.0000
hsa04520	Adherens junction	60	72	5	10	2	0.5317	0.3675
hsa04540	Gap junction	37	36	5	6	1	0.4568	0.1813
hsa04550	Signaling pathways regulating pluripotency of stem cells	64	58	9	4	3	0.3052	0.1044
ko02024	Quorum sensing	238	230	35	13	1	0.3805	1154
ko05111	Biofilm formation - Vibrio cholerae	52	64	3	13	1	0.4765	0.0833
ko02025	Biofilm formation - Pseudomonas aeruginosa	47	47	3	9	1	0.1875	0.0104
ko02026	Biofilm formation - Escherichia coli	66	74	6	11	1	0.3125	0.1045
ko02030	Bacterial chemotaxis	28	31	1	7	1	0.0003	1.86E-07
hsa04810	Regulation of actin cytoskeleton	70	81	2	10	1	0.5643	0.0625
hsa04610	Complement and coagulation cascades	47	58	3	10	1	0.1875	0.0313
hsa04611	Platelet activation	77	84	4	5	4	0.5743	0.2188
hsa04620	Toll-like receptor signaling pathway	75	96	3	11	1	0.4732	0.1250
dme04624	Toll and Imd signaling pathway	59	60	7	7	1	0.4312	0.1140
hsa04621	NOD-like receptor signaling pathway	154	160	17	12	2	0.6843	0.3840
hsa04622	RIG-I-like receptor signaling pathway	49	59	5	12	1	0.7632	0.3000
hsa04623	Cytosolic DNA-sensing pathway	29	34	2	6	1	0.2862	0.0156
hsa04625	C-type lectin receptor signaling pathway	183	156	37	6	3	0.6396	0.3180
hsa04650	Natural killer cell mediated cytotoxicity	76	98	9	8	2	0.6525	0.3056
hsa04612	Antigen processing and presentation	27	22	5	3	5	0.4437	0.1600
hsa04658	Th1 and Th2 cell differentiation	61	113	2	12	1	0.4623	0.0625

Table A.16 continues: Summary of the results of pathway analysis

KEGG ID	Name	Number of nodes	Number of edges	Components	Max Connectivity	Number of nodes with maximum connectivity	$Rscore$	$PRscore$
hsa04659	Th17 cell differentiation	81	82	15	11	1	0.6417	0.286
hsa04657	IL-17 signaling pathway	9	8	1	3	1	0.1718	0.0039
hsa04662	B cell receptor signaling pathway	44	43	7	7	1	0.4643	0.2679
hsa04664	Fc epsilon RI signaling pathway	34	38	3	8	1	0.5962	0.2083
hsa04666	Fc gamma R-mediated phagocytosis	54	70	2	8	1	0.6712	0.1250
hsa04670	Leukocyte transendothelial migration	55	48	7	4	3	0.8736	0.6790
hsa04672	Intestinal immune network for IgA production	29	19	12	4	1	0.8164	0.6276
hsa04911	Insulin secretion	38	38	3	6	1	0.4127	0.1250
hsa04910	Insulin signaling pathway	64	66	4	8	1	0.5624	0.1328
hsa04922	Glucagon signaling pathway	46	49	3	10	1	0.3671	0.1250
hsa04923	Regulation of lipolysis in adipocytes	44	38	7	4	2	0.5562	0.3943
hsa04920	Adipocytokine signaling pathway	31	38	1	6	1	0.0016	0.0000
hsa03320	PPAR signaling pathway	60	65	6	27	1	0.5872	0.2292
hsa04912	GnRH signaling pathway	31	26	6	3	5	0.4624	0.1719
hsa04913	Ovarian steroidogenesis	30	25	6	5	2	0.5503	0.2712
hsa04915	Estrogen signaling pathway	69	67	4	6	1	0.0005	1.39E-04
xla04914	Progesterone-mediated oocyte maturation	32	22	10	3	2	0.6483	0.3377
hsa04917	Prolactin signaling pathway	52	55	3	16	1	0.2836	0.0625
hsa04921	Oxytocin signaling pathway	65	73	2	8	1	0.2643	0.0313
hsa04926	Relaxin signaling pathway	89	102	4	8	1	0.0488	0.0020
hsa04918	Thyroid hormone synthesis	36	36	7	5	2	0.8240	0.6786
hsa04919	Thyroid hormone signaling pathway	73	71	8	10	2	0.7453	0.5938
hsa04928	Parathyroid hormone synthesis, secretion and action	69	66	5	10	1	0.3284	0.0789
hsa04916	Melanogenesis	33	27	6	4	1	0.3911	0.1361
hsa04924	Renin secretion	38	35	3	5	1	0.0856	0.0039
hsa04614	Renin-angiotensin system	13	10	5	3	2	0.6500	0.3375
hsa04927	Cortisol synthesis and secretion	28	42	2	7	1	0.5000	0.1250
hsa04260	Cardiac muscle contraction	14	16	2	3	8	0.3750	0.1880
hsa04261	Adrenergic signaling in cardiomyocytes	58	68	2	13	1	0.0061	1.53E-05
hsa04270	Vascular smooth muscle contraction	62	61	6	6	1	0.5936	0.2917
hsa04970	Salivary secretion	27	24	3	3	2	0.2206	0.0231
hsa04971	Gastric acid secretion	28	25	3	4	1	0.1715	0.0109
hsa04972	Pancreatic secretion	26	23	3	3	5	0.3022	0.0261
hsa04976	Bile secretion	21	18	3	5	1	0.3072	6.77E-02
hsa04973	Carbohydrate digestion and absorption	11	9	2	2	7	0.3125	0.0234
hsa04979	Cholesterol metabolism	41	26	15	3	1	0.7958	0.5794

Table A.16 continues: Summary of the results of pathway analysis

KEGG ID	Name	Number of nodes	Number of edges	Components	Max Connectivity	Number of nodes with maximum connectivity	$Rscore$	$PRscore$
hsa04978	Mineral absorption	22	14	8	6	1	0.8769	0.7227
hsa04962	Vasopressin-regulated water re-absorption	11	10	2	4	1	0.5712	9.1270
hsa04960	Aldosterone-regulated sodium re-absorption	19	20	1	7	1	0.3237	0.0000
hsa04961	Endocrine and other factor-regulated calcium re-absorption	40	32	10	5	1	0.7712	0.6758
hsa04966	Collecting duct acid secretion	10	6	4	2	2	0.5000	0.3178
hsa04724	Glutamatergic synapse	44	51	1	11	1	0.1327	0.0000
hsa04727	GABAergic synapse	23	19	4	4	2	0.7518	0.5625
hsa04725	Cholinergic synapse	39	45	1	10	1	0.1683	0.0000
hsa04728	Dopaminergic synapse	42	54	1	8	1	0.1952	0.0000
hsa04726	Serotonergic synapse	44	40	8	6	1	0.4617	0.2070
hsa04720	Long-term potentiation	23	28	1	5	1	0.0016	0.0000
hsa04730	Long-term depression	26	23	4	4	1	0.3366	0.0938
hsa04721	Synaptic vesicle cycle	13	8	5	3	1	1.0000	0.8500
hsa04722	Neurotrophin signaling pathway	77	111	4	17	1	0.1243	0.0488
hsa04744	Phototransduction	29	20	9	3	2	0.5902	0.4201
dme04745	Phototransduction - fly	36	35	4	5	2	0.5043	0.2501
hsa04740	Olfactory transduction	31	28	5	4	4	0.3347	0.563
hsa04742	Taste transduction	45	48	10	5	5	0.4140	0.1078
hsa04750	Inflammatory mediator regulation of TRP channels	81	80	7	9	1	0.3752	0.1540
dme04320	Dorso-ventral axis formation	18	16	2	4	1	0.3762	0.0624
hsa04360	Axon guidance	128	153	12	7	2	0.2838	0.1660
hsa04380	Osteoclast differentiation	67	79	2	12	1	0.4312	0.0625
hsa04211	Longevity regulating pathway	48	54	1	8	3	7.11E-12	1.78E-13
cel04212	Longevity regulating pathway - worm	57	59	9	9	1	0.3181	0.1329
hsa04213	Longevity regulating pathway - multiple species	49	52	5	6	1	0.3040	0.1251
hsa04713	Circadian entrainment	41	42	4	5	1	0.3701	0.1406
ath04712	Circadian rhythm - plant	31	38	3	8	1	0.4166	0.2210
hsa04714	Thermogenesis	54	65	2	6	2	0.4876	0.0625
ath04626	Plant-pathogen interaction	31	29	6	5	2	0.5890	0.2500
hsa05200	Pathways in cancer	159	169	19	10	2	0.6721	0.2795
hsa05230	Central carbon metabolism in cancer	31	31	4	9	1	0.4731	0.2188
hsa05231	Choline metabolism in cancer	37	41	3	5	1	0.5000	0.2083
hsa05202	Transcriptional misregulation in cancer	16	9	7	3	1	0.9107	0.6964
hsa05206	MicroRNAs in cancer	342	230	115	7	1	0.5288	0.3493
hsa05210	Colorectal cancer	49	45	6	6	1	0.3516	0.1354
hsa05212	Pancreatic cancer	51	44	9	4	1	0.5003	0.2847
hsa05225	Hepatocellular carcinoma	77	66	13	8	1	0.5144	0.3032
hsa05226	Gastric cancer	80	66	15	6	1	0.5033	0.2385

Table A.16 continues: Summary of the results of pathway analysis

KEGG ID	Name	Number of nodes	Number of edges	Components	Max Connectivity	Number of nodes with maximum connectivity	$Rscore$	$PRscore$
hsa05214	Glioma	58	67	5	5	7	0.4527	0.0813
hsa05216	Thyroid cancer	16	12	4	3	2	0.5312	0.1250
hsa05221	Acute myeloid leukemia	35	42	2	7	2	0.0625	0.0039
hsa05220	Chronic myeloid leukemia	36	33	4	5	1	0.2355	0.0696
hsa05217	Basal cell carcinoma	16	13	3	5	1	0.5190	0.2087
hsa05218	Melanoma	19	18	2	3	5	0.3110	0.0524
hsa05211	Renal cell carcinoma	39	33	7	8	1	0.5103	0.1409
hsa05219	Bladder cancer	14	10	4	3	1	0.5000	0.1641
hsa05215	Prostate cancer	38	37	3	11	1	0.2285	0.0244
hsa05213	Endometrial cancer	21	20	2	5	1	0.1767	0.0212
hsa05224	Breast cancer	103	98	16	6	3	0.5834	0.4619
hsa05222	Small cell lung cancer	26	25	3	7	1	0.3457	0.0635
hsa05223	Non-small cell lung cancer	34	35	4	5	2	0.1433	0.0469
hsa05322	Systemic lupus erythematosus	12	7	5	2	2	0.7000	0.3500
hsa05323	Rheumatoid arthritis	13	7	6	2	1	0.8125	0.5000
hsa05320	Autoimmune thyroid disease	14	7	7	1	14	0.7500	0.4464
hsa05321	Inflammatory bowel disease (IBD)	53	54	6	5	2	0.3181	0.1147
hsa05330	Allograft rejection	14	7	7	1	14	0.7857	0.4643
hsa05010	Alzheimer disease	34	25	9	3	4	0.4444	0.2118
hsa05012	Parkinson disease	22	16	6	3	2	0.7604	0.5521
hsa05014	Amyotrophic lateral sclerosis (ALS)	38	35	8	5	1	0.4567	0.2207
hsa05016	Huntington disease	15	11	4	3	1	0.5253	0.1914
hsa05020	Prion diseases	26	19	7	6	1	0.4877	0.2176
hsa05030	Cocaine addiction	43	41	4	4	3	0.2060	0.0314
hsa05031	Amphetamine addiction	60	51	12	5	1	0.4577	0.2510
hsa05032	Morphine addiction	41	36	8	5	1	0.3779	0.0825
hsa05033	Nicotine addiction	30	17	13	2	4	0.7500	0.4808
hsa05034	Alcoholism	49	45	7	4	3	0.5452	0.3259
hsa05410	Hypertrophic cardiomyopathy (HCM)	17	19	3	4	3	1.0000	0.7500
hsa05412	Arrhythmogenic right ventricular cardiomyopathy (ARVC)	14	8	6	2	2	0.8750	0.6667
hsa05414	Dilated cardiomyopathy (DCM)	27	28	4	4	4	0.7624	0.5630
hsa05416	Viral myocarditis	21	15	6	2	9	0.6510	0.3125
hsa04930	Type II diabetes mellitus	14	14	2	8	1	0.2507	0.0317
hsa04950	Maturity onset diabetes of the young	27	29	3	6	1	0.2403	0.0234
hsa04932	Non-alcoholic fatty liver disease (NAFLD)	62	62	6	7	1	0.6582	0.2501
hsa04931	Insulin resistance	99	114	8	11	1	0.2954	0.1450
hsa04933	AGE-RAGE signaling pathway in diabetic complications	64	93	3	19	2	0.1873	0.0260
hsa04934	Cushing syndrome	61	74	7	7	1	0.3088	0.1164
hsa05110	Vibrio cholerae infection	17	10	7	3	1	0.7053	0.3571

Table A.16 continues: Summary of the results of pathway analysis

KEGG ID	Name	Number of nodes	Number of edges	Components	Max Connectivity	Number of nodes with maximum connectivity	$Rscore$	$PRscore$
hsa05120	Epithelial cell signaling in Helicobacter pylori infection	26	21	7	4	1	0.5993	0.3037
hsa05130	Pathogenic Escherichia coli infection	14	10	4	3	1	0.5000	0.2031
hsa05131	Shigellosis	24	24	3	5	2	0.4270	0.0902
hsa05133	Pertussis	31	28	7	4	3	0.4980	0.2411
hsa05134	Legionellosis	30	25	7	9	1	0.5358	0.3082
hsa05150	Staphylococcus aureus infection	24	18	6	5	1	0.6691	0.4584
hsa05152	Tuberculosis	114	136	13	12	1	0.4872	0.2093
hsa05100	Bacterial invasion of epithelial cells	37	34	4	3	7	0.3134	0.0430
hsa05166	Human T-cell leukemia virus 1 infection	107	101	23	11	1	0.7429	0.3778
hsa05170	Human immunodeficiency virus 1 infection	107	102	12	7	1	0.4183	0.2124
hsa05162	Measles	64	59	11	5	4	0.5599	0.3092
hsa05164	Influenza A	79	85	10	10	2	0.6705	0.4125
hsa05161	Hepatitis B	100	85	15	6	1	0.3773	0.1460
hsa05160	Hepatitis C	51	45	6	4	4	0.3014	0.0938
hsa05168	Herpes simplex infection	70	68	9	7	2	0.5773	0.2370
hsa05163	Human cytomegalovirus infection	140	131	17	5	1	0.3039	0.0998
hsa05167	Kaposi sarcoma-associated herpesvirus infection	100	102	11	6	1	0.2873	0.0814
hsa05169	Epstein-Barr virus infection	69	80	9	12	2	0.5878	0.3687
hsa05165	Human papillomavirus	115	99	20	9	1	0.4966	0.2629
hsa05146	Amoebiasis	19	12	7	2	5	0.6428	0.3571
hsa05144	Malaria	10	6	4	2	2	0.8125	0.5000
hsa05145	Toxoplasmosis	53	47	7	6	1	0.3778	0.1563
hsa05140	Leishmaniasis	42	43	2	5	3	0.0045	4.29E-04
hsa05142	Chagas disease (American trypanosomiasis)	63	60	10	11	1	0.5587	0.2257
hsa05143	African trypanosomiasis	17	11	6	3	1	0.6850	0.3385
ko01501	beta-Lactam resistance	56	64	4	24	1	0.0878	0.0630
ko01503	Cationic antimicrobial peptide (CAMP) resistance	92	78	27	11	1	0.7823	0.6019
hsa01521	EGFR tyrosine kinase inhibitor resistance	62	121	1	14	5	4.34E-15	1.08E-18
hsa01524	Platinum drug resistance	34	41	2	8	1	0.5000	0.1250
hsa01523	Antifolate resistance	11	8	3	4	1	0.3541	0.1250
hsa01522	Endocrine resistance	64	85	6	8	1	0.4388	0.1462

Table A.17: Summary of drop in stability for each of the 26 groups of the pathways

Group name	$Rscore$	$PRscore$	Drop in stability (%)
Genetic information processing	0.5807	0.3026	47.89
Signal transduction	0.3858	0.1660	56.96
Signaling molecules and interaction	0.7175	0.4259	40.64
Transport and catabolism	0.6034	0.3364	44.25
Cell growth and death	0.4175	0.2137	48.81
Cellular community - eukaryotes	0.3275	0.1633	50.13
Cellular community - prokaryotes	0.3392	0.0784	76.89
Cell motility	0.5198	0.2147	58.69
Endocrine system	0.4134	0.1918	53.60
Circulatory system	0.3249	0.1597	50.83
Digestive system	0.4266	0.2076	51.34
Excretory system	0.5415	0.2801	48.27
Nervous system	0.3524	0.1957	44.45
Sensory system	0.4436	0.1976	55.44
Development	0.3637	0.0804	77.89
Aging	0.3550	0.1331	62.49
Cancers: Overview	0.6333	0.3504	44.66
Cancers: Specific types	0.4470	0.1955	56.27
Neurodegenerative diseases	0.5349	0.2787	47.89
Substance dependence	0.6250	0.3848	38.42
Endocrine and metabolic diseases	0.3234	0.0987	69.46
Infectious diseases: Bacterial	0.5478	0.2460	55.09
Infectious diseases: Viral	0.4870	0.2365	51.43
Infectious diseases: Parasitic	0.5135	0.2629	48.80
Drug resistance: Antimicrobial	0.4350	0.3324	23.58
Drug resistance: Antineoplastic	0.3463	0.0990	71.40

Appendix B

File formats

B.1 The FASTA file format

FASTA format, used in Chapters 3 and 5, is a text-based format for representing either nucleotide sequences or peptide sequences, in which base pairs or amino acids are represented using single-letter codes. A sequence in FASTA format begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater-than (">") symbol. The description line contains the name of genes (nucleotide sequence) and proteins (amino acid sequence). The next line holds the sequence formed by the single letter codes. A FASTA file is a series of such FASTA sequences and are usually stored in the form of *filename.fasta*.

Example of a nucleotide sequence in FASTA format:

>E2F transcription factor 4 (N)

ATGGCGGAGGCCGGGCCACAGGCGCCGCCGCCCCCGGGTACTCCAAGCC ...

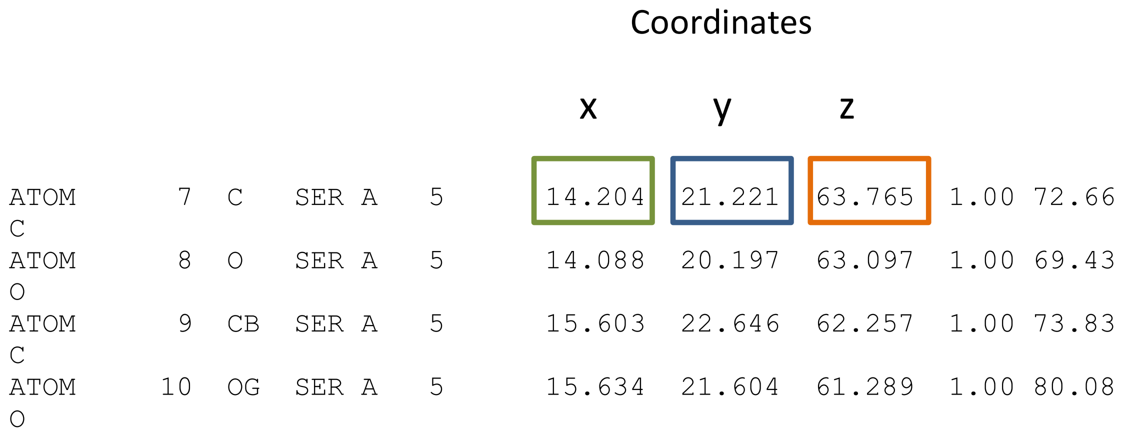
Example of an amino acid sequence in FASTA format:

>MCHU - Calmodulin - Human, rabbit, bovine, rat, and chicken

ADQLTEEQIAEFKEAFSLFDKDGDGTITTKELGTVMRSLGQGTEAFSLTTKQIAT...

B.2 The PDB file format

The Protein Data Bank (PDB) [37] format, used in Chapter 4, provides a standard representation for macromolecular structure data derived from X-ray diffraction and NMR studies. It is a textual file format describing the three-dimensional structures of molecules stored in the Protein Data Bank. The pdb format accordingly provides the description and annotation of protein and nucleic acid structures including atomic coordinates, secondary structure assignments, as well as atomic connectivity. Additionally, experimental metadata are stored. A pdb file for a certain protein contains such structural information in them and are usually stored in the form of *filename.pdb*. A snapshot of the file format (the part of the file that was mainly used in the thesis) has been given in Figure B.1.


Coordinates
x
y
z

```
ATOM      7  C    SER A   5      14.204  21.221  63.765  1.00 72.66
C
ATOM      8  O    SER A   5      14.088  20.197  63.097  1.00 69.43
O
ATOM      9  CB   SER A   5      15.603  22.646  62.257  1.00 73.83
C
ATOM     10  OG   SER A   5      15.634  21.604  61.289  1.00 80.08
O
```

Figure B.1: A snapshot of a PDB file. The coordinates of every atom in a molecule is given by the columns annotated by x (7th column), y (8th column), z (9th column).

B.3 The KCF file format

The KEGG Chemical Format (KCF) [209] files, used in Chapter 6, gives the description of the chemical structure of metabolites in a two-dimensional (2D) format, where each metabolite can be represented as a graph consisting of vertices (atoms) and edges (bonds). KCF files are stored as *filename.txt*. Example of a KCF file given in Figure B.2.

B.4 The KGML file format

The KEGG Markup Language (KGML) [209], the input format of the algorithm developed in Chapter 7, is an exchange format of the KEGG pathway maps. KGML enables automatic drawing of KEGG pathways, and facilitates computational analysis and modeling of gene/protein and chemical networks. The KGML files for metabolic pathway maps contain

```
ENTRY       C00009                      Compound
ATOM        5
            1   P1b P    27.1282  -21.2572
            2   O1c O    25.9971  -21.9011
            3   O1c O    28.4489  -21.8360
            4   O1c O    27.4637  -22.7728
            5   O1c O    27.1282  -19.9529
BOND        4
            1     1    2 1
            2     1    3 1
            3     1    4 1
            4     1    5 2
///
```

Figure B.2: A snapshot of a KCF file. The "Entry" section contains the unique compound ID which helps in uniquely identifying each compound in KEGG. The "Atom" section contains the list of atoms (3rd column) present in the metabolite. The "Bond" section contains the list of bonds (4th column and 5th column being the two atoms between which a bond exists) among these atoms.

Protein descriptors:

```
<entry id="335" name="hsa:353500 hsa:656" type="gene"
        link="https://www.kegg.jp/dbget-
bin/www_bget?hsa:353500+hsa:656">
        <graphics name="BMP8A..." fgcolor="#000000" bgcolor="#BFFFBF"
             type="rectangle" x="1661" y="669" width="46"
height="17"/>
    </entry>
```

Interaction between two proteins:

```
    <relation entry1="335" entry2="437" type="PPrel">
        <subtype name="activation" value="--&gt;"/>
    </relation>
```

Figure B.3: A snapshot of a KGML file. The "Protein descriptors" section lists all the proteins involved in a particular pathway and assigns a unique ID to the protein involved in the pathway. Each protein description is given under the XML tag "entry". The "Interaction between two proteins" section lists interactions between various proteins. These interactions are defined under the tag "relation".

two types of graph object patterns - how boxes (enzymes) are linked by "relations" and how circles (chemical compounds) are linked by "reactions". The KGML files for non-metabolic pathway maps contain only the aspect of how boxes (proteins) are linked by "relations". A KGML file for a certain pathway is stored in the form of *filename.xml*. Example of a KGML file is given in Figure B.3.

Bibliography

[1] A. M. Abdallah, N. C. G. Van Pittius, P. A. D. Champion, J. Cox, J. Luirink, C. M. Vandenbroucke-Grauls, B. J. Appelmelk, and W. Bitter, "Type VII secretion-mycobacteria show the way," *Nature Reviews Microbiology*, vol. 5, no. 11, pp. 883–891, 2007.

[2] A. M. Abdallah, T. Verboom, E. M. Weerdenburg, N. C. Gey van Pittius, P. W. Mahasha, C. Jiménez, M. Parra, N. Cadieux, M. J. Brennan, B. J. Appelmelk *et al.*, "PPE and PE_PGRS proteins of Mycobacterium marinum are transported via the Type VII secretion system ESX-5," *Molecular Microbiology*, vol. 73, no. 3, pp. 329–340, 2009.

[3] I. Abubakar, P. Gautret, G. W. Brunette, L. Blumberg, D. Johnson, G. Poumerol, Z. A. Memish, M. Barbeschi, and A. S. Khan, "Global perspectives for prevention of infectious diseases associated with mass gatherings," *The Lancet Infectious Diseases*, vol. 12, no. 1, pp. 66–74, 2012.

[4] T. Akutsu, S. Kosub, A. A. Melkman, and T. Tamura, "Finding a periodic attractor of a boolean network," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 5, pp. 1410–1421, 2012.

[5] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano, "A system for identifying genetic networks from gene expression patterns produced by gene disruptions and overexpressions," *Genome Informatics*, vol. 9, pp. 151–160, 1998.

[6] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano, "Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions," in *9th ACM-SIAM Symposium on Discrete Algorithms*, vol. 98. Citeseer, San Francisco, CA, USA, January, 1998, pp. 695–702.

[7] M. Ala'raj and M. F. Abbod, "Classifiers consensus system approach for credit scoring," *Knowledge-Based Systems*, vol. 104, pp. 89–105, 2016.

[8] P. Albersheim and A. J. Anderson, "Proteins from plant cell walls inhibit polygalacturonases secreted by plant pathogens," *Proceedings of the National Academy of Sciences*, vol. 68, no. 8, pp. 1815–1819, 1971.

[9] P. Albersheim and B. S. Valent, "Host-pathogen interactions VII. Plant pathogens secrete proteins which inhibit enzymes of the host capable of attacking the pathogen," *Plant Physiology*, vol. 53, no. 5, pp. 684–687, 1974.

[10] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, "Introduction to pathogens," in *Molecular Biology of the Cell. 4th edition.* Garland Science, 2002.

[11] L. J. Alderwick, L. G. Dover, M. Seidel, R. Gande, H. Sahm, L. Eggeling, and G. S. Besra, "Arabinan-deficient mutants of Corynebacterium glutamicum and the consequent flux in decaprenylmonophosphoryl-D-arabinose metabolism," *Glycobiology*, vol. 16, no. 11, pp. 1073–1081, 2006.

[12] S. Alguwaizani, B. Park, X. Zhou, D.-S. Huang, and K. Han, "Predicting interactions between virus and host proteins using repeat patterns and composition of amino acids," *Journal of Healthcare Engineering*, vol. 2018, 2018.

[13] Y. An, J. Wang, C. Li, A. Leier, T. Marquez-Lago, J. Wilksch, Y. Zhang, G. I. Webb, J. Song, and T. Lithgow, "Comprehensive assessment and performance improvement of effector protein predictors for bacterial secretion systems III, IV and VI," *Briefings in bioinformatics*, vol. 19, no. 1, pp. 148–161, 2018.

[14] Y. An, J. Wang, C. Li, J. Revote, Y. Zhang, T. Naderer, M. Hayashida, T. Akutsu, G. I. Webb, T. Lithgow *et al.*, "SecretEPDB: a comprehensive web-based resource for secreted effector proteins of the bacterial Types III, IV and VI secretion systems," *Scientific Reports*, vol. 7, no. 41031, pp. 1–10, 2017.

[15] J. P. F. Angeli, M. Schneider, B. Proneth, Y. Y. Tyurina, V. A. Tyurin, V. J. Hammond, N. Herbach, M. Aichler, A. Walch, E. Eggenhofer *et al.*, "Inactivation of the ferroptosis regulator gpx4 triggers acute renal failure in mice," *Nature Cell Biology*, vol. 16, no. 12, pp. 1180–1191, 2014.

[16] P. Argos, J. M. Rao, and P. A. Hargrave, "Structural prediction of membrane-bound proteins," *European Journal of Biochemistry*, vol. 128, no. 2-3, pp. 565–575, 1982.

[17] R. Arnold, S. Brandmaier, F. Kleine, P. Tischler, E. Heinz, S. Behrens, A. Niinikoski, H.-W. Mewes, M. Horn, and T. Rattei, "Sequence-based prediction of Type III secreted proteins," *PLoS Pathogens*, vol. 5, no. 4, p. e1000376, 2009.

[18] M.-S. Aschtgen, C. S. Bernard, S. De Bentzmann, R. Lloubes, and E. Cascales, "SciN is an outer membrane lipoprotein required for Type VI secretion in enteroaggregative Escherichia coli," *Journal of Bacteriology*, vol. 190, no. 22, pp. 7523–7531, 2008.

[19] T. K. Attwood, "The babel of bioinformatics," *Science*, vol. 290, no. 5491, pp. 471–473, 2000.

[20] D. F. Aubert, H. Xu, J. Yang, X. Shi, W. Gao, L. Li, F. Bisaro, S. Chen, M. A. Valvano, and F. Shao, "A burkholderia Type VI effector deamidates Rho GTPases to activate the pyrin inflammasome and trigger inflammation," *Cell Host & Microbe*, vol. 19, no. 5, pp. 664–674, 2016.

[21] F. Audibert, M. Jolivet, L. Chedid, R. t. Arnon, and M. Sela, "Successful immunization with a totally synthetic diphtheria vaccine," *Proceedings of the National Academy of Sciences*, vol. 79, no. 16, pp. 5042–5046, 1982.

[22] G. J. Augustine, M. E. Burns, W. M. DeBello, D. L. Pettit, and F. E. Schweizer, "Exocytosis: proteins and perturbations," *Annual Review of Pharmacology and Toxicology*, vol. 36, no. 1, pp. 659–701, 1996.

[23] C. Aurrecoechea, A. Barreto, J. Brestelli, B. P. Brunk, S. Cade, R. Doherty, S. Fischer, B. Gajria, X. Gao, A. Gingle *et al.*, "EuPathDB: the eukaryotic pathogen database," *Nucleic Acids Research*, vol. 41, no. D1, pp. D684–D691, 2013.

[24] C. Aurrecoechea, J. Brestelli, B. P. Brunk, S. Fischer, B. Gajria, X. Gao, A. Gingle, G. Grant, O. S. Harb, M. Heiges *et al.*, "EuPathDB: a portal to eukaryotic pathogen databases," *Nucleic Acids Research*, vol. 38, no. suppl 1, pp. D415–D419, 2010.

[25] T. Baba and O. Schneewind, "Instruments of microbial warfare: bacteriocin synthesis, toxicity and immunity," *Trends in Microbiology*, vol. 6, no. 2, pp. 66–71, 1998.

[26] D. Baker and A. Sali, "Protein structure prediction and structural genomics," *Science*, vol. 294, no. 5540, pp. 93–96, 2001.

[27] S. Balakrishnan, O. Tastan, J. Carbonell, and J. Klein-Seetharaman, "Alternative paths in HIV-1 targeted human signal transduction pathways," *BMC Genomics*, vol. 10, no. 3, pp. 1–13, 2009.

[28] P. Baldi, S. Brunak, and F. Bach, *Bioinformatics: the Machine Learning Approach*. London, England:MIT press, 2001.

[29] P. Baldi and P. Sadowski, "The dropout learning algorithm," *Artificial intelligence*, vol. 210, pp. 78–122, 2014.

[30] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, 1996.

[31] R. M. Barkin, "Diphtheria pertussis-tetanus vaccine: Reactogenicity," *Pediatrics*, vol. 63, no. 2, pp. 256–260, 1979.

[32] M. Basler, "Type VI secretion system: secretion by a contractile nanomachine," *Philosophical Transactions of the Royal Society B*, vol. 370, no. 1679, pp. 1–11, 2015.

[33] V. Bellotti, P. Mangione, and G. Merlini, "immunoglobulin light chain amyloidosis—the archetype of structural and pathogenic variability," *Journal of Structural Biology*, vol. 130, no. 2-3, pp. 280–289, 2000.

[34] A. Ben-David, "About the relationship between ROC curves and cohen's kappa," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 6, pp. 874–882, 2008.

[35] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler, "Genbank," *Nucleic Acids Research*, vol. 36, no. Database issue, p. D25, 2008.

[36] H. Berman, K. Henrick, H. Nakamura, and J. L. Markley, "The worldwide protein data bank (wwPDB): ensuring a single, uniform archive of PDB data," *Nucleic Acids Research*, vol. 35, no. 1, pp. 301–303, 2006.

[37] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic Acids Research*, vol. 28, no. 1, pp. 235–242, 2000.

[38] A. Bertoletti, M. K. Maini, and C. Ferrari, "The host-pathogen interaction during HBV infection: immunological controversies," *Antiviral Therapy*, vol. 15, no. 3, pp. 15–24, 2010.

[39] R. Bhaskaran and P. Ponnuswamy, "Positional flexibilities of amino acid residues in globular proteins," *International Journal of Peptide and Protein Research*, vol. 32, no. 4, pp. 241–255, 1988.

[40] D. Bi, L. Liu, C. Tai, Z. Deng, K. Rajakumar, and H.-Y. Ou, "SecReT4: a web-based bacterial Type IV secretion system resource," *Nucleic Acids Research*, vol. 41, no. D1, pp. D660–D665, 2012.

[41] C. C. Bigelow, "On the average hydrophobicity of proteins and the relation between it and protein structure," *Journal of Theoretical Biology*, vol. 16, no. 2, pp. 187–211, 1967.

[42] S. Bleves, I. Dunger, M. C. Walter, D. Frangoulidis, G. Kastenmüller, R. Voulhoux, and A. Ruepp, "HoPaCI-DB: host-Pseudomonas and Coxiella interaction database," *Nucleic Acids Research*, vol. 42, no. D1, pp. D671–D676, 2014.

[43] T. L. Blundell, B. L. Sibanda, R. W. Montalvão, S. Brewerton, V. Chelliah, C. L. Worth, N. J. Harmer, O. Davies, and D. Burke, "Structural biology and bioinformatics in drug design: opportunities and challenges for target identification and lead discovery," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 361, no. 1467, pp. 413–423, 2006.

[44] J. R. Bock and D. A. Gough, "Predicting protein– protein interactions from primary structure," *Bioinformatics*, vol. 17, no. 5, pp. 455– 460, 2001.

[45] K. Bonde, "The genus Datura: From research subject to powerful hallucinogen," *Ethnobotanical Leaflets*, vol. 1998, no. 1, pp. 1–8, 1998.

[46] T. Bose, C. Das, A. Dutta, V. Mahamkali, S. Sadhu, and S. S. Mande, "Understanding the role of interactions between host and Mycobacterium tuberculosis under hypoxic condition: an in silico approach," *BMC genomics*, vol. 19, no. 1, pp. 1–13, 2018.

[47] T. Bose, K. Venkatesh, and S. S. Mande, "Computational analysis of host–pathogen protein interactions between humans and different strains of enterohemorrhagic Escherichia coli," *Frontiers in Cellular and Infection Microbiology*, vol. 7, no. 128, pp. 1–14, 2017.

[48] H. Botella, G. Stadthagen, G. Lugo-Villarino, C. de Chastellier, and O. Neyrolles, "Metallobiology of host–pathogen interactions: an intoxicating new insight," *Trends in Microbiology*, vol. 20, no. 3, pp. 106–112, 2012.

[49] L. S. Boutemy, S. R. King, J. Win, R. K. Hughes, T. A. Clarke, T. M. Blumenschein, S. Kamoun, and M. J. Banfield, "Structures of Phytophthora RXLR effector proteins a conserved but adaptable fold underpins functional diversity," *Journal of Biological Chemistry*, vol. 286, no. 41, pp. 35 834–35 842, 2011.

[50] A. L. Brass, D. M. Dykxhoorn, Y. Benita, N. Yan, A. Engelman, R. J. Xavier, J. Lieberman, and S. J. Elledge, "Identification of host proteins required for HIV infection through a functional genomic screen," *Science*, vol. 319, no. 5865, pp. 921–926, 2008.

[51] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[52] J. M. Breitenbach and R. P. Hausinger, “Proteus mirabilis urease. Partial purification and inhibition by boric acid and boronic acids,” *Biochemical Journal*, vol. 250, no. 3, pp. 917–920, 1988.

[53] K. E. Broglie, P. Biddle, R. Cressman, and R. Broglie, “Functional analysis of DNA sequences responsible for ethylene regulation of a bean chitinase gene in transgenic tobacco.” *The Plant Cell*, vol. 1, no. 6, pp. 599–607, 1989.

[54] Y. R. Brunet, A. Zoued, F. Boyer, B. Douzi, and E. Cascales, “The Type VI secretion TssEFGK-VgrG phage-like baseplate is recruited to the tssjlm membrane complex via multiple contacts and serves as assembly platform for tail tube/sheath polymerization,” *PLoS Genetics*, vol. 11, no. 10, p. e1005545, 2015.

[55] H. B. Bull and K. Breese, “Surface tension of amino acid solutions: a hydrophobicity scale of the amino acid residues,” *Archives of Biochemistry and Biophysics*, vol. 161, no. 2, pp. 665–670, 1974.

[56] D. Bumann, “Heterogeneous host-pathogen encounters: act locally, think globally,” *Cell Host & Microbe*, vol. 17, no. 1, pp. 13–19, 2015.

[57] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, “Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem,” *Advances in Knowledge Discovery and Data Mining*, pp. 475–482, 2009.

[58] B. J. Burkinshaw, G. Prehna, L. J. Worrall, and N. C. Strynadka, “Structure of salmonella effector protein SopB N-terminal domain in complex with host Rho GTPase Cdc42,” *Journal of Biological Chemistry*, vol. 287, no. 16, pp. 13 348–13 355, 2012.

[59] D. Burstein, T. Zusman, E. Degtyar, R. Viner, G. Segal, and T. Pupko, “Genome-scale identification of Legionella pneumophila effectors using a machine learning approach,” *PLoS Pathogens*, vol. 5, no. 7, p. e1000508, 2009.

[60] M. L. Burts, W. A. Williams, K. DeBord, and D. M. Missiakas, “EsxA and EsxB are secreted by an ESAT-6-like system that is required for the pathogenesis of Staphylococcus aureus infections,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 4, pp. 1169–1174, 2005.

[61] M. A. Calderwood, K. Venkatesan, L. Xing, M. R. Chase, A. Vazquez, A. M. Holthaus, A. E. Ewence, N. Li, T. Hirozane-Kishikawa, D. E. Hill *et al.*, “Epstein–Barr virus and virus human protein interaction maps,” *Proceedings of the National Academy of Sciences*, vol. 104, no. 18, pp. 7606–7611, 2007.

[62] A. M. Campbell and L. J. Heyer, *Discovering Genomics, Proteomics, and Bioinformatics*. New Delhi, India:Dorling Kindersley, 2003, no. QH447 C35 2007.

[63] J. Y. Cao and S. J. Dixon, “Mechanisms of ferroptosis,” *Cellular and Molecular Life Sciences*, vol. 73, no. 11-12, pp. 2195–2209, 2016.

[64] E. Capriotti, R. Calabrese, and R. Casadio, “Predicting the insurgence of human genetic diseases associated to single point protein mutations with support vector machines and evolutionary information,” *Bioinformatics*, vol. 22, no. 22, pp. 2729–2734, 2006.

[65] M. Carsiotis, B. Stocker, D. Weinstein, and A. O’Brien, “A Salmonella typhimurium virulence gene linked to flg.” *Infection and Immunity*, vol. 57, no. 11, pp. 3276–3280, 1989.

[66] A. Casadevall and L.-a. Pirofski, “Host-pathogen interactions: redefining the basic concepts of virulence and pathogenicity,” *Infection and Immunity*, vol. 67, no. 8, pp. 3703–3713, 1999.

[67] A. Casadevall and L.-a. Pirofski, “Host-pathogen interactions: basic concepts of microbial commensalism, colonization, infection, and disease,” *Infection and Immunity*, vol. 68, no. 12, pp. 6511–6518, 2000.

[68] A. Casadevall and L.-a. Pirofski, “Host-pathogen interactions: the attributes of virulence,” *The Journal of Infectious Diseases*, vol. 184, no. 3, pp. 337–344, 2001.

[69] E. Cascales, “The Type VI secretion toolkit,” *EMBO Reports*, vol. 9, no. 8, pp. 735–741, 2008.

[70] E. Cascales and P. J. Christie, “The versatile bacterial Type IV secretion systems,” *Nature Reviews Microbiology*, vol. 1, no. 2, pp. 137–149, 2003.

[71] C.-Y. Chang, M.-T. Hsu, E. X. Esposito, and Y. J. Tseng, “Oversampling to overcome overfitting: exploring the relationship between data set composition, molecular descriptors, and predictive modeling methods,” *Journal of Chemical Information and Modeling*, vol. 53, no. 4, pp. 958–971, 2013.

[72] J. H. Chang, D. Desveaux, and A. L. Creason, “The abcs and 123s of bacterial secretion systems in plant pathogenesis,” *Annual Review of Phytopathology*, vol. 52, pp. 317–345, 2014.

[73] M. Charton, “Protein folding and the genetic code: an alternative quantitative model,” *Journal of Theoretical Biology*, vol. 91, no. 1, pp. 115–123, 1981.

[74] M. Charton and B. I. Charton, “The structural dependence of amino acid hydrophobicity parameters,” *Journal of Theoretical Biology*, vol. 99, no. 4, pp. 629–644, 1982.

[75] A. Chatr-aryamontri, A. Ceol, D. Peluso, A. Nardozza, S. Panni, F. Sacco, M. Tinti, A. Smolyar, L. Castagnoli, M. Vidal *et al.*, “VirusMINT: a viral protein interaction database,” *Nucleic Acids Research*, vol. 37, no. 1, pp. D669–D673, 2009.

[76] D. Chaussabel, R. T. Semnani, M. A. McDowell, D. Sacks, A. Sher, and T. B. Nutman, “Unique gene expression profiles of human macrophages and dendritic cells to phylogenetically distinct parasites,” *Blood*, vol. 102, no. 2, pp. 672–681, 2003.

[77] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[78] J. Chen, N. Sawyer, and L. Regan, “Protein–protein interactions: general trends in the relationship between binding affinity and interfacial buried surface area,” *Protein Science*, vol. 22, no. 4, pp. 510–515, 2013.

[79] L. Chen, Z. Xiong, L. Sun, J. Yang, and Q. Jin, “Vfdb 2012 update: toward the genetic diversity and molecular evolution of bacterial virulence factors,” *Nucleic Acids Research*, vol. 40, no. D1, pp. D641–D645, 2012.

[80] L. Chen, J. Yang, J. Yu, Z. Yao, L. Sun, Y. Shen, and Q. Jin, “VFDB: a reference database for bacterial virulence factors,” *Nucleic Acids Research*, vol. 33, no. 1, pp. D325–D328, 2005.

[81] Z. Chen, P. Zhao, F. Li, A. Leier, T. T. Marquez-Lago, Y. Wang, G. I. Webb, A. I. Smith, R. J. Daly, K.-C. Chou *et al.*, “iFeature: a python package and web server for features extraction and selection from protein and peptide sequences,” *Bioinformatics*, vol. 34, no. 14, pp. 2499–2502, 2018.

[82] E. Chertova, O. Chertov, L. V. Coren, J. D. Roser, C. M. Trubey, J. W. Bess, R. C. Sowder, E. Barsov, B. L. Hood, R. J. Fisher *et al.*, “Proteomic and biochemical analysis of purified human immunodeficiency virus type 1 produced from infected monocyte-derived macrophages,” *Journal of Virology*, vol. 80, no. 18, pp. 9039–9052, 2006.

[83] S.-M. Choo and K.-H. Cho, “An efficient algorithm for identifying primary phenotype attractors of a large-scale boolean network,” *BMC Systems Biology*, vol. 10, no. 1, pp. 95–108, 2016.

[84] C. Chothia, “Structural invariants in protein folding,” *Nature*, vol. 254, no. 5498, pp. 304–308, 1975.

[85] C. Chothia, “The nature of the accessible and buried surfaces in proteins,” *Journal of Molecular Biology*, vol. 105, no. 1, pp. 1–12, 1976.

[86] D. L. Clemens, B.-Y. Lee, and M. A. Horwitz, “Purification, characterization, and genetic analysis of Mycobacterium tuberculosis urease, a potentially critical determinant of host-pathogen interaction,” *Journal of Bacteriology*, vol. 177, no. 19, pp. 5644–5652, 1995.

[87] S. Cole, R. Brosch, J. Parkhill, T. Garnier, C. Churcher, D. Harris, S. Gordon, K. Eiglmeier, S. Gas, C. r. Barry *et al.*, “Deciphering the biology of Mycobacterium tuberculosis from the complete genome sequence,” *Nature*, vol. 393, no. 6685, pp. 537–544, 1998.

[88] S. R. Collins, P. Kemmeren, X.-C. Zhao, J. F. Greenblatt, F. Spencer, F. C. Holstege, J. S. Weissman, and N. J. Krogan, “Toward a comprehensive atlas of the physical interactome of Saccharomyces cerevisiae,” *Molecular & Cellular Proteomics*, vol. 6, no. 3, pp. 439–450, 2007.

[89] M. L. Connolly, “Solvent-accessible surfaces of proteins and nucleic acids,” *Science*, vol. 221, no. 4612, pp. 709–713, 1983.

[90] U. Consortium, “UniProt: a hub for protein information,” *Nucleic Acids Research*, vol. 43, no. 1, pp. 204–212, 2015.

[91] S. E. Converse and J. S. Cox, “A protein secretion pathway critical for Mycobacterium tuberculosis virulence is conserved and functional in Mycobacterium smegmatis,” *Journal of Bacteriology*, vol. 187, no. 4, pp. 1238–1245, 2005.

[92] N. R. Coordinators, “Database resources of the national center for biotechnology information,” *Nucleic Acids Research*, vol. 45, no. 1, pp. 13–21, 2017.

[93] T. R. Costa, C. Felisberto-Rodrigues, A. Meir, M. S. Prevost, A. Redzej, M. Trokter, and G. Waksman, “Secretion systems in Gram-negative bacteria: structural and mechanistic insights,” *Nature Reviews Microbiology*, vol. 13, no. 6, pp. 343–359, 2015.

[94] K. Dalal, “Counting the onion,” *Random Structures & Algorithms*, vol. 24, no. 2, pp. 155–165, 2004.

[95] T. Dallas, A. W. Park, and J. M. Drake, “Predicting cryptic links in host-parasite networks,” *PLoS Computational Biology*, vol. 13, no. 5, p. e1005557, 2017.

[96] M. Datt, “Geometric analysis of the conformational features of protein structures,” in *BIOMAT 2015: Proceedings of the International Symposium on Mathematical and*

Computational Biology. World Scientific, Roorkee, UT, India, November 1-7, 2016, p. 166.

[97] M. I. Davidich and S. Bornholdt, "Boolean network model predicts cell cycle sequence of fission yeast," *PloS One*, vol. 3, no. 2, p. e1672, 2008.

[98] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2, pp. 224–227, 1979.

[99] F. P. Davis, D. T. Barkan, N. Eswar, J. H. McKerrow, and A. Sali, "Host–pathogen protein interactions predicted by comparative modeling," *Protein Science*, vol. 16, no. 12, pp. 2585–2596, 2007.

[100] W. K. Dawson, M. Maciejczyk, E. J. Jankowska, and J. M. Bujnicki, "Coarse-grained modeling of RNA 3D structure," *Methods*, vol. 103, pp. 138–156, 2016.

[101] M. Dayhoff, R. Schwartz, and B. Orcutt, "A model of evolutionary change in proteins," *Atlas of Protein Sequence and Structure*, vol. 5, pp. 345–352, 1978.

[102] R. de AB Assis, L. C. Polloni, J. S. Patané, S. Thakur, É. B. Felestrino, J. Diaz-Caballero, L. A. Digiampietri, L. R. Goulart, N. F. Almeida, R. Nascimento *et al.*, "Identification and analysis of seven effector protein families with different adaptive and evolutionary histories in plant-associated members of the xanthomonadaceae," *Scientific Reports*, vol. 7, no. 1, pp. 1–17, 2017.

[103] H. K. de Jong, C. M. Parry, T. van der Poll, and W. J. Wiersinga, "Host–pathogen interaction in invasive salmonellosis," *PLOS Pathogen*, vol. 8, no. 10, p. e1002933, 2012.

[104] M. I. De Jonge, G. Pehau-Arnaudet, M. M. Fretz, F. Romain, D. Bottai, P. Brodin, N. Honoré, G. Marchal, W. Jiskoot, P. England *et al.*, "ESAT-6 from Mycobacterium tuberculosis dissociates from its putative chaperone cfp-10 under acidic conditions and exhibits membrane-lysing activity," *Journal of Bacteriology*, vol. 189, no. 16, pp. 6028–6034, 2007.

[105] P. Dean, "Functional domains and motifs of bacterial type III effector proteins and their roles in infection," *FEMS Microbiology Reviews*, vol. 35, no. 6, pp. 1100–1125, 2011.

[106] K. C. Dee, D. A. Puleo, and R. Bizios, *An introduction to tissue-biomaterial interactions*. Hoboken, NJ, USA:John Wiley & Sons, 2003.

[107] R. DeLa Cadena, K. J. Laskin, R. A. Pixley, R. B. Sartor, J. H. Schwab, N. Back, G. S. Bedi, R. S. Fisher, and R. W. Colman, "Role of kallikrein-kinin system in pathogenesis of bacterial cell wall-induced inflammation," *American Journal of Physiology-Gastrointestinal and Liver Physiology*, vol. 260, no. 2, pp. 213–219, 1991.

[108] G. Delogu and M. J. Brennan, "Comparative immune response to PE and PE_PGRS antigens of Mycobacterium tuberculosis," *Infection and Immunity*, vol. 69, no. 9, pp. 5606–5611, 2001.

[109] C. d'Enfert, A. Ryter, and A. Pugsley, "Cloning and expression in Escherichia coli of the Klebsiella pneumoniae genes for production, surface localization and secretion of the lipoprotein pullulanase." *The EMBO Journal*, vol. 6, no. 11, pp. 3531–3538, 1987.

[110] S. Depluverez, S. Devos, and B. Devreese, "The role of bacterial secretion systems in the virulence of gram-negative airway pathogens associated with cystic fibrosis," *Frontiers in Microbiology*, vol. 7, p. 1336, 2016.

[111] S. C. Derrick and S. L. Morris, "The ESAT6 protein of Mycobacterium tuberculosis induces apoptosis of macrophages by activating caspase expression," *Cellular Microbiology*, vol. 9, no. 6, pp. 1547–1555, 2007.

[112] V. Devloo, P. Hansen, and M. Labbé, "Identification of all steady states in large networks by logical analysis," *Bulletin of Mathematical Biology*, vol. 65, no. 6, pp. 1025–1051, 2003.

[113] R. Díaz-Uriarte and S. A. De Andres, "Gene selection and classification of microarray data using random forest," *BMC Bioinformatics*, vol. 7, no. 1, pp. 1–13, 2006.

[114] J. Dickson, R. Syamananda, and A. Flangas, "The genetic approach to the physiology of parasitism of the corn rust pathogens," *American Journal of Botany*, pp. 614–620, 1959.

[115] V. J. DiRita, "Co-ordinate expression of virulence genes by ToxR in Vibrio cholerae," *Molecular Microbiology*, vol. 6, no. 4, pp. 451–458, 1992.

[116] C. P. Doherty, "Host-pathogen interactions: the role of iron," *The Journal of Nutrition*, vol. 137, no. 5, pp. 1341–1344, 2007.

[117] J. M. Doolittle and S. M. Gomez, "Structural similarity-based predictions of protein interactions between HIV-1 and Homo sapiens," *Virology Journal*, vol. 7, no. 1, pp. 1–15, 2010.

[118] J. M. Doolittle and S. M. Gomez, “Mapping protein interactions between Dengue virus and its human and insect hosts,” *PLoS Neglected Tropical Disease*, vol. 5, no. 2, p. e954, 2011.

[119] T. Driscoll, M. D. Dyer, T. Murali, and B. W. Sobral, “PIG-the pathogen interaction gateway,” *Nucleic Acids Research*, vol. 37, no. 1, pp. D647–D650, 2009.

[120] D. Dua and C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[121] E. Dubrova, M. Teslenko, and A. Martinelli, “Kauffman networks: Analysis and applications,” in *ICCAD-2005. IEEE/ACM International Conference on Computer-Aided Design*. IEEE Computer Society, San Jose, CA, USA, November 6-10, 2005, pp. 479–484.

[122] E. Durand, C. Cambillau, E. Cascales, and L. Journet, “Vgrg, tae, tle, and beyond: the versatile arsenal of Type VI secretion effectors,” *Trends in Microbiology*, vol. 22, no. 9, pp. 498–507, 2014.

[123] S. Durmuş, T. Çakır, A. Özgür, and R. Guthke, “A review on computational systems biology of pathogen–host interactions,” *Frontiers in Microbiology*, vol. 6, pp. 235–253, 2015.

[124] S. Durmuş Tekir, T. Çakır, E. Ardıç, A. S. Sayılırbaş, G. Konuk, M. Konuk, H. Sarıyer, A. Uğurlu, İ. Karadeniz, A. Özgür *et al.*, “PHISTO: pathogen–host interaction search tool,” *Bioinformatics*, vol. 29, no. 10, pp. 1357–1358, 2013.

[125] M. D. Dyer, T. Murali, and B. W. Sobral, “Computational prediction of host-pathogen protein–protein interactions,” *Bioinformatics*, vol. 23, no. 13, pp. i159–i166, 2007.

[126] M. D. Dyer, T. Murali, and B. W. Sobral, “The landscape of human proteins interacting with viruses and other pathogens,” *PLoS Pathogen*, vol. 4, no. 2, p. e32, 2008.

[127] M. D. Dyer, C. Neff, M. Dufford, C. G. Rivera, D. Shattuck, J. Bassaganya-Riera, T. Murali, and B. W. Sobral, “The human-bacterial pathogen protein interaction networks of Bacillus anthracis, Francisella tularensis, and Yersinia pestis,” *PloS One*, vol. 5, no. 8, p. e12089, 2010.

[128] H. Edwards, P. Allen *et al.*, “A fine-structure study of the primary infection process during infection of barley by Erysiphe graminis f. sp. hordei,” *Phytopathology*, vol. 60, no. 10, pp. 1504–1509, 1970.

[129] B. Efron, “The convex hull of a random set of points,” *Biometrika*, vol. 52, no. 3-4, pp. 331–343, 1965.

[130] D. Eisenberg and A. D. McLachlan, “Solvation energy in protein folding and binding,” *Nature*, vol. 319, no. 6050, pp. 199–203, 1986.

[131] D. Eisenberg, R. M. Weiss, T. C. Terwilliger, and W. Wilcox, “Hydrophobic moments and protein structure,” vol. 17, pp. 109–120, 1982.

[132] L. B. Ellis, J. Gao, K. Fenner, and L. P. Wackett, “The university of Minnesota pathway prediction system: predicting metabolic logic,” *Nucleic Acids Research*, vol. 36, no. suppl 2, pp. W427–W432, 2008.

[133] E. Emmenegger, E. Kentop, T. Thompson, S. Pittam, A. Ryan, D. Keon, J. Carlino, J. Ranson, R. Life, R. Troyer *et al.*, “Development of an aquatic pathogen database (Aquapathogen X) and its utilization in tracking emerging fish virus pathogens in North America,” *Journal of Fish Diseases*, vol. 34, no. 8, pp. 579–587, 2011.

[134] P. D. English and P. Albersheim, “Host-pathogen interactions: I. A Correlation between α-galactosidase production and virulence,” *Plant Physiology*, vol. 44, no. 2, pp. 217–224, 1969.

[135] R. Espíndola and N. Ebecken, “On extending f-measure and g-mean metrics to multiclass problems,” *WIT Transactions on Information and Communication Technologies*, vol. 35, 2005.

[136] P. Evans, W. Dampier, L. Ungar, and A. Tozeren, “Prediction of HIV-1 virus-host protein interactions using virus and host sequence motifs,” *BMC Medical Genomics*, vol. 2, no. 1, pp. 1–13, 2009.

[137] M. E. Fahey, M. J. Bennett, C. Mahon, S. Jäger, L. Pache, D. Kumar, A. Shapiro, K. Rao, S. K. Chanda, C. S. Craik *et al.*, “GPS-Prot: a web-based visualization platform for integrating host-pathogen interaction data,” *BMC Bioinformatics*, vol. 12, no. 1, pp. 1–3, 2011.

[138] C. Farrow, J. Heidel, J. Maloney, and J. Rogers, “Scalar equations for synchronous boolean networks with biological applications,” *IEEE Transactions on Neural Networks*, vol. 15, no. 2, pp. 348–354, 2004.

[139] G. D. Fasman, *Practical handbook of Biochemistry and Molecular Biology*. Cleveland, OH, USA: CRC press, 1989.

[140] J.-L. FAUCHERE, M. Charton, L. B. Kier, A. Verloop, and V. Pliska, “Amino acid side chain parameters for correlation studies in biology and pharmacology,” *International Journal of Peptide and Protein Research*, vol. 32, no. 4, pp. 269–278, 1988.

[141] G. Fenhalls, L. Stevens, L. Moses, J. Bezuidenhout, J. C. Betts, P. van Helden, P. T. Lukey, and K. Duncan, “In situ detection of Mycobacterium tuberculosis transcripts in human lung granulomas reveals differential gene expression in necrotic lesions,” *Infection and Immunity*, vol. 70, no. 11, pp. 6330–6338, 2002.

[142] R. E. Ferguson, H. P. Carroll, A. Harris, E. R. Maher, P. J. Selby, and R. E. Banks, “Housekeeping proteins: a preliminary study illustrating some limitations as useful references in protein expression studies,” *Proteomics*, vol. 5, no. 2, pp. 566–571, 2005.

[143] M. L. Fisher, A. J. Anderson, and P. Albersheim, “Host-pathogen interactions VI. A single plant protein efficiently inhibits endopolygalacturonases secreted by Colletotrichum lindemuthianum and Aspergillus niger,” *Plant Physiology*, vol. 51, no. 3, pp. 489–491, 1973.

[144] J. L. Fleiss and J. Cohen, “The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability,” *Educational and Psychological Measurement*, vol. 33, no. 3, pp. 613–619, 1973.

[145] Å. Flobak, A. Baudot, E. Remy, L. Thommesen, D. Thieffry, M. Kuiper, and A. Lægreid, “Discovery of drug synergies in gastric cancer cells predicted by logical modeling,” *PLoS Computational Biology*, vol. 11, no. 8, p. e1004426, 2015.

[146] H. F. Fumia and M. L. Martins, “Boolean network model for cancer pathways: predicting carcinogenesis and targeted therapy outcomes,” *PloS One*, vol. 8, no. 7, p. e69008, 2013.

[147] J. E. Galan and R. Curtiss, “Cloning and molecular characterization of genes whose products allow Salmonella typhimurium to penetrate tissue culture cells,” *Proceedings of the National Academy of Sciences*, vol. 86, no. 16, pp. 6383–6387, 1989.

[148] J. E. Galán, M. Lara-Tejero, T. C. Marlovits, and S. Wagner, “Bacterial Type III secretion systems: specialized nanomachines for protein delivery into target cells,” *Annual Review of Microbiology*, vol. 68, pp. 415–438, 2014.

[149] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, “An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes,” *Pattern Recognition*, vol. 44, no. 8, pp. 1761–1776, 2011.

[150] V. Ganganwar, "An overview of classification algorithms for imbalanced datasets," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 4, pp. 42–47, 2012.

[151] J. Gao, L. B. Ellis, and L. P. Wackett, "The university of Minnesota biocatalysis/biodegradation database: improving public access," *Nucleic Acids Research*, vol. 38, no. suppl 1, pp. D488–D491, 2010.

[152] J. P. Garel, D. Filliol, and P. Mandel, "Coefficients de partage d'aminoacides, nucleobases, nucleosides et nucleotides dans un systeme solvant salin," *Journal of Chromatography A*, vol. 78, no. 2, pp. 381–391, 1973.

[153] K. Georgopoulou, D. Smirlis, S. Bisti, E. Xingi, L. Skaltsounis, and K. Soteriadou, "In vitro activity of 10-deacetylbaccatin III against leishmania donovani promastigotes and intracellular amastigotes," *Planta Medica*, vol. 73, no. 10, pp. 1081–1088, 2007.

[154] A. Ghosh, B. C. Dhara, and R. K. De, "Comparative analysis of cluster validity indices in identifying some possible genes mediating certain cancers," *Molecular Informatics*, vol. 32, no. 4, pp. 347–354, 2013.

[155] Z. Ghosh, B. Mallick, and J. Chakrabarti, "Cellular versus viral microRNAs in host–virus interaction," *Nucleic Acids Research*, vol. 37, no. 4, pp. 1035–1048, 2009.

[156] D. Goldsack and R. Chalifoux, "Contribution of the free energy of mixing of hydrophobic side chains to the stability of the tertiary structure of proteins," *Journal of Theoretical Biology*, vol. 39, no. 3, pp. 645–651, 1973.

[157] E. Gómez-Díaz, M. Jordà, M. A. Peinado, and A. Rivero, "Epigenetics of host–pathogen interactions: the road ahead and the road behind," *PLoS Pathogen*, vol. 8, no. 11, p. e1003007, 2012.

[158] N. F. Goodacre, D. L. Gerloff, and P. Uetz, "Protein domains of unknown function are essential in bacteria," *mBio*, vol. 5, no. 1, pp. e00 744–13, 2014.

[159] B. D. Grant and J. G. Donaldson, "Pathways and mechanisms of endocytic recycling," *Nature Reviews: Molecular Cell Biology*, vol. 10, no. 9, pp. 597–608, 2009.

[160] M. Griffiths, *Understanding Pathogen Behaviour: Virulence, Stress Response and Resistance*. Abington Hall, AB, Cambridge:Elsevier, 2005.

[161] E. J. Groen and T. H. Gillingwater, "UBA1: at the crossroads of ubiquitin homeostasis and neurodegeneration," *Trends in Molecular Medicine*, vol. 21, no. 10, pp. 622–632, 2015.

[162] W.-j. Guan, W.-h. Liang, Y. Zhao, H.-r. Liang, Z.-s. Chen, Y.-m. Li, X.-q. Liu, R.-c. Chen, C.-l. Tang, T. Wang *et al.*, “Comorbidity and its impact on 1590 patients with COVID–19 in china: A nationwide analysis,” *European Respiratory Journal*, vol. 55, no. 5, pp. 1–24, 2020.

[163] I. Guérin and C. de Chastellier, “Pathogenic mycobacteria disrupt the macrophage actin filament network,” *Infection and Immunity*, vol. 68, no. 5, pp. 2655–2662, 2000.

[164] K. M. Guinn, M. J. Hickey, S. K. Mathur, K. L. Zakel, J. E. Grotzke, D. M. Lewinsohn, S. Smith, and D. R. Sherman, “Individual RD1-region genes are required for export of ESAT-6/CFP-10 and for virulence of Mycobacterium tuberculosis,” *Molecular Microbiology*, vol. 51, no. 2, pp. 359–370, 2004.

[165] T. Guirimand, S. Delmotte, and V. Navratil, “VirHostNet 2.0: surfing on the web of virus/host molecular interactions data,” *Nucleic Acids Research*, vol. 43, no. D1, pp. D583–D587, 2015.

[166] R. A. Günster, S. A. Matthews, D. W. Holden, and T. L. Thurston, “SseK1 and SseK3 Type III secretion system effectors inhibit NF-κb signaling and necroptotic cell death in salmonella-infected macrophages,” *Infection and Immunity*, vol. 85, no. 3, pp. e00 010–17, 2017.

[167] Z. S. Guo, A. Naik, M. E. O’Malley, P. Popovic, R. Demarco, Y. Hu, X. Yin, S. Yang, H. J. Zeh, B. Moss *et al.*, “The enhanced tumor selectivity of an oncolytic vaccinia lacking the host range and antiapoptosis genes SPI-1 and SPI-2,” *Cancer Research*, vol. 65, no. 21, pp. 9991–9998, 2005.

[168] S. Gupta, S. S. Bisht, R. Kukreti, S. Jain, and S. K. Brahmachari, “Boolean network analysis of a neurotransmitter signaling pathway,” *Journal of Theoretical Biology*, vol. 244, no. 3, pp. 463–469, 2007.

[169] A. Gur and Y. Cohen, “The peach replant problem-some causal agents,” *Soil Biology and Biochemistry*, vol. 21, no. 6, pp. 829–834, 1989.

[170] M. G. Gutierrez, S. S. Master, S. B. Singh, G. A. Taylor, M. I. Colombo, and V. Deretic, “Autophagy is a defense mechanism inhibiting BCG and Mycobacterium tuberculosis survival in infected macrophages,” *Cell*, vol. 119, no. 6, pp. 753–766, 2004.

[171] A. H Meijer and H. P Spaink, “Host-pathogen interactions made transparent with the zebrafish model,” *Current Drug Targets*, vol. 12, no. 7, pp. 1000–1017, 2011.

[172] J. Haiko and B. Westerlund-Wikström, “The role of the bacterial flagellum in adhesion and virulence,” *Biology*, vol. 2, no. 4, pp. 1242–1267, 2013.

[173] D. Haller, L. Holt, S. C. Kim, R. F. Schwabe, R. B. Sartor, and C. Jobin, “Transforming growth factor-β1 inhibits non-pathogenic gramnegative bacteria-induced NF-κb recruitment to the interleukin-6 gene promoter in intestinal epithelial cells through modulation of histone acetylation,” *Journal of Biological Chemistry*, vol. 278, no. 26, pp. 23 851–23 860, 2003.

[174] R. Hamid, M. A. Khan, M. Ahmad, M. M. Ahmad, M. Z. Abdin, J. Musarrat, S. Javed *et al.*, “Chitinases: an update,” *Journal of Pharmacy and Bioallied Sciences*, vol. 5, no. 1, pp. 21–29, 2013.

[175] H. Han, W.-Y. Wang, and B.-H. Mao, “Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning,” in *International Conference on Intelligent Computing*. Springer, Hefei, Anhui, China, August 23-26, 2005, pp. 878–887.

[176] Y. Han, D. Zhou, X. Pang, Y. Song, L. Zhang, J. Bao, Z. Tong, J. Wang, Z. Guo, J. Zhai *et al.*, “Microarray Analysis of Temperature-Induced Transcriptome of Yersinia pestis,” *Microbiology and Immunology*, vol. 48, no. 11, pp. 791–805, 2004.

[177] J. A. Hanley and B. J. McNeil, “The meaning and use of the area under a receiver operating characteristic (ROC) curve.” *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[178] K. Hara, D. Saitoh, and H. Shouno, “Analysis of dropout learning regarded as ensemble learning,” in *International Conference on Artificial Neural Networks*. Springer, Barcelona, Spain, September 6-9, 2016, pp. 72–79.

[179] A. Harvey, K. Bradley, S. Cochran, E. Rowan, J. Pratt, J. Quillfeldt, and D. Jerusalinsky, “What can toxins tell us for drug discovery?” *Toxicon*, vol. 36, no. 11, pp. 1635–1640, 1998.

[180] S. K. Hatzios, S. Abel, J. Martell, T. Hubbard, J. Sasabe, D. Munera, L. Clark, D. A. Bachovchin, F. Qadri, E. T. Ryan *et al.*, “Chemoproteomic profiling of host and pathogen enzymes active in cholera,” *Nature Chemical Biology*, vol. 12, no. 4, pp. 268–274, 2016.

[181] G. He, H. Han, and W. Wang, “An over-sampling expert system for learing from imbalanced data sets,” in *Neural Networks and Brain*, Beijing, China, October 13-15, 2005.

[182] H. Hegyi and M. Gerstein, "The relationship between protein structure and function: a comprehensive survey with application to the yeast genome 1," *Journal of Molecular Biology*, vol. 288, no. 1, pp. 147–164, 1999.

[183] A. Heinken, M. T. Khan, G. Paglia, D. A. Rodionov, H. J. Harmsen, and I. Thiele, "Functional metabolic map of Faecalibacterium prausnitzii, a beneficial human gut microbe," *Journal of Bacteriology*, vol. 196, no. 18, pp. 3289–3302, 2014.

[184] C. Helene, "Intermolecular interactions and biomolecular organization," *Trends in Biochemical Sciences*, vol. 2, no. 12, pp. 289–290, 1977.

[185] S. Helisalmi, M. Hiltunen, A. Mannermaa, A. M. Koivisto, M. Lehtovirta, I. Alafuzoff, M. Ryynänen, and H. Soininen, "Is the presenilin-1 E318G missense mutation a risk factor for Alzheimer's disease?" *Neuroscience Letters*, vol. 278, no. 1-2, pp. 65–68, 2000.

[186] S. R. Heller, A. McNaught, I. Pletnev, S. Stein, and D. Tchekhovskoi, "InChI, the IUPAC international chemical identifier," *Journal of Cheminformatics*, vol. 7, no. 1, pp. 1–23, 2015.

[187] S. Hess and A. Rambukkana, "Bacterial-induced cell reprogramming to stem cell-like cells: new premise in host–pathogen interactions," *Current Opinion in Microbiology*, vol. 23, pp. 179–188, 2015.

[188] W. Hess, "Ultrastructure of onion roots infected with Pyrenochaeta terrestris, a fungus parasite," *American Journal of Botany*, pp. 832–845, 1969.

[189] A. Hoffmann, A. Levchenko, M. L. Scott, and D. Baltimore, "The IκB-NF-κB signaling module: temporal control and selective gene activation," *Science*, vol. 298, no. 5596, pp. 1241–1245, 2002.

[190] T. Hsu, S. M. Hingley-Wilson, B. Chen, M. Chen, A. Z. Dai, P. M. Morin, C. B. Marks, J. Padiyar, C. Goulding, M. Gingery *et al.*, "The primary mechanism of attenuation of bacillus calmette-guerin is a loss of secreted lytic function required for invasion of lung interstitial tissue," *Proceedings of the National Academy of Sciences*, vol. 100, no. 21, pp. 12 420–12 425, 2003.

[191] J. Hu, X. He, D.-J. Yu, X.-B. Yang, J.-Y. Yang, and H.-B. Shen, "A new supervised over-sampling algorithm with application to protein-nucleotide binding residue prediction," *PLoS One*, vol. 9, no. 9, p. e107676, 2014.

[192] S. Huang, "Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery," *Journal of Molecular Medicine*, vol. 77, no. 6, pp. 469–480, 1999.

[193] S. Huang, I. Ernberg, and S. Kauffman, "Cancer attractors: a systems view of tumors from a gene network dynamics and developmental perspective," vol. 20, no. 7, pp. 869–876, September 2009.

[194] M. Huerta, G. Downing, F. Haseltine, B. Seto, and Y. Liu, "NIH working definition of bioinformatics and computational biology," *US National Institute of Health*, pp. 1–1, 2000.

[195] H. S. Hussein and J. M. Brasel, "Toxicity, metabolism, and impact of mycotoxins on humans and animals," *Toxicology*, vol. 167, no. 2, pp. 101–134, 2001.

[196] J. O. Hutchens and H. Sober, "Heat capacities, absolute entropies, and entropies of formation of amino acids and related compounds," *Handbook of Biochemistry*, pp. B60–B61, 1970.

[197] J. E. Irazoqui, A. Ng, R. J. Xavier, and F. M. Ausubel, "Role for β-catenin and HOX transcription factors in Caenorhabditis elegans and mammalian host epithelial-pathogen interactions," *Proceedings of the National Academy of Sciences*, vol. 105, no. 45, pp. 17 469–17 474, 2008.

[198] H. D. Isenberg, "Pathogenicity and virulence: another view." *Clinical Microbiology Reviews*, vol. 1, no. 1, pp. 40–53, 1988.

[199] T. S. Istivan and P. J. Coloe, "Phospholipase A in Gram-negative bacteria and its role in pathogenesis," *Microbiology*, vol. 152, no. 5, pp. 1263–1274, 2006.

[200] J. Janin, S. Wodak, M. Levitt, and B. Maigret, "Conformation of amino acid side-chains in proteins," *Journal of Molecular Biology*, vol. 125, no. 3, pp. 357–386, 1978.

[201] D. D. Jones, "Amino acid properties and side-chain orientation in proteins: a cross correlation approach," *Journal of Theoretical Biology*, vol. 50, no. 1, pp. 167–183, 1975.

[202] K. Jong, J. Mary, A. Cornuéjols, E. Marchiori, and M. Sebag, "Ensemble feature ranking," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, Pisa, Italy, September 20-24, 2004, pp. 267–278.

[203] A. P. Junqueira-Kipnis, R. J. Basaraba, V. Gruppo, G. Palanisamy, O. C. Turner, T. Hsu, W. R. Jacobs Jr, S. A. Fulton, S. M. Reba, W. H. Boom *et al.*, "Mycobacteria

lacking the RD1 region do not induce necrosis in the lungs of mice lacking interferon-γ," *Immunology*, vol. 119, no. 2, pp. 224–231, 2006.

[204] G. Jurman, S. Riccadonna, and C. Furlanello, "A comparison of MCC and CEN error measures in multi-class prediction," *PLoS One*, vol. 7, no. 8, p. e41882, 2012.

[205] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features," *Biopolymers*, vol. 22, no. 12, pp. 2577–2637, 1983.

[206] C. Kaimer and P. L. Graumann, "Players between the worlds: multifunctional DNA translocases," *Current Opinion in Microbiology*, vol. 14, no. 6, pp. 719–725, 2011.

[207] S. Kakraba, D. Knisley *et al.*, "A graph-theoretic model of single point mutations in the cystic fibrosis transmembrane conductance regulator," *Journal of Advances in Biotechnology*, vol. 6, no. 1, pp. 780–786, 2016.

[208] S. Kanamaru, "Structural similarity of tailed phages and pathogenic bacterial secretion systems," *Proceedings of the National Academy of Sciences*, vol. 106, no. 11, pp. 4067–4068, 2009.

[209] M. Kanehisa and S. Goto, "KEGG: kyoto encyclopedia of genes and genomes," *Nucleic Acids Research*, vol. 28, no. 1, pp. 27–30, 2000.

[210] A. Karlyshev, P. Oyston, K. Williams, G. Clark, R. Titball, E. Winzeler, and B. Wren, "Application of High-Density Array-Based Signature-Tagged Mutagenesis To Discover Novel Yersinia Virulence-Associated Genes," *Infection and Immunity*, vol. 69, no. 12, pp. 7810–7819, 2001.

[211] P. D. Karp, C. A. Ouzounis, C. Moore-Kochlacs, L. Goldovsky, P. Kaipa, D. Ahrén, S. Tsoka, N. Darzentas, V. Kunin, and N. López-Bigas, "Expansion of the BioCyc collection of pathway-genome databases to 160 genomes," *Nucleic Acids Research*, vol. 33, no. 19, pp. 6083–6089, 2005.

[212] P. D. Karp, S. Paley, and P. Romero, "The pathway tools software," *Bioinformatics*, vol. 18, no. suppl 1, pp. S225–S232, 2002.

[213] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437–467, 1969.

[214] B. Kearney, P. C. Ronald, D. Dahlbeck, and B. J. Staskawicz, "Molecular basis for evasion of plant host defence in bacterial spot disease of pepper," *Nature*, vol. 332, no. 6164, pp. 541–543, 1988.

[215] K. Khoufache, O. Puel, N. Loiseau, M. Delaforge, D. Rivollet, A. Coste, C. Cordonnier, E. Escudier, F. Botterel, and S. Bretagne, "Verruculogen associated with aspergillus fumigatus hyphae and conidia modifies the electrophysiological properties of human nasal epithelial cells," *BMC Microbiology*, vol. 7, no. 1, pp. 1–11, 2007.

[216] F. Kierszenbaum, J. J. Wirth, P. P. McCann, and A. Sjoerdsma, "Impairment of macrophage function by inhibitors of ornithine decarboxylase activity." *Infection and Immunity*, vol. 55, no. 10, pp. 2461–2464, 1987.

[217] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[218] M. W. Kirschner, "The meaning of systems biology," *Cell*, vol. 121, no. 4, pp. 503–504, 2005.

[219] J. Klein, A. Kumar, and M. McKennon, "Phospholipase D inhibitors and uses thereof," 2004, US Patent App. 10/405,059.

[220] S. Kmiecik, D. Gront, M. Kolinski, L. Wieteska, A. E. Dawid, and A. Kolinski, "Coarse-grained protein models and their applications," *Chemical Reviews*, vol. 116, no. 14, pp. 7898–7936, 2016.

[221] L. A. Knodler, J. A. Ibarra, E. Pérez-Rueda, C. K. Yip, and O. Steele-Mortimer, "Coiled-coil domains enhance the membrane association of Salmonella type III effectors," *Cellular Microbiology*, vol. 13, no. 10, pp. 1497–1517, 2011.

[222] R. Kohavi and D. Sommerfield, "Feature subset selection using the wrapper method: Overfitting and dynamic search space topology," pp. 192–197, 1995.

[223] R. Kohavi, D. Sommerfield, and J. Dougherty, "Data mining using a machine learning library in C++," *International Journal on Artificial Intelligence Tools*, vol. 6, no. 04, pp. 537–566, 1997.

[224] P. Kohl, E. J. Crampin, T. Quinn, and D. Noble, "Systems biology: an approach," *Clinical Pharmacology & Therapeutics*, vol. 88, no. 1, pp. 25–33, 2010.

[225] R. Kolodny, D. Petrey, and B. Honig, "Protein structure comparison: implications for the nature of 'fold space', and structure and function prediction," *Current Opinion in Structural Biology*, vol. 16, no. 3, pp. 393–398, 2006.

[226] R. König, S. Stertz, Y. Zhou, A. Inoue, H.-H. Hoffmann, S. Bhattacharyya, J. G. Alamares, D. M. Tscherne, M. B. Ortigoza, Y. Liang *et al.*, "Human host factors

required for influenza virus replication," *Nature*, vol. 463, no. 7282, pp. 813–817, 2010.

[227] R. König, Y. Zhou, D. Elleder, T. L. Diamond, G. M. Bonamy, J. T. Irelan, C.-y. Chiang, B. P. Tu, P. D. De Jesus, C. E. Lilley *et al.*, "Global analysis of host-pathogen interactions that regulate early-stage HIV-1 replication," *Cell*, vol. 135, no. 1, pp. 49–60, 2008.

[228] A. M. Krachler, H. Ham, and K. Orth, "Outer membrane adhesion factor multivalent adhesion molecule 7 initiates host cell binding during infection by gram-negative pathogens," *Proceedings of the National Academy of Sciences*, vol. 108, no. 28, pp. 11 614–11 619, 2011.

[229] O. Krishnadev and N. Srinivasan, "A data integration approach to predict host-pathogen protein-protein interactions: application to recognize protein interactions between human and a malarial parasite," *In Silico Biology*, vol. 8, no. 3, 4, pp. 235–250, 2008.

[230] M. Kshirsagar, J. Carbonell, and J. Klein-Seetharaman, "Techniques to cope with missing data in host–pathogen protein interaction prediction," *Bioinformatics*, vol. 28, no. 18, pp. i466–i472, 2012.

[231] M. Kshirsagar, J. Carbonell, and J. Klein-Seetharaman, "Multitask learning for host–pathogen protein interactions," *Bioinformatics*, vol. 29, no. 13, pp. i217–i226, 2013.

[232] M. Kubat, S. Matwin *et al.*, "Addressing the curse of imbalanced training sets: one-sided selection," in *International Conference on Machine Learning*, vol. 97, Nashville, TN, USA, July 8-12, 1997, pp. 179–186.

[233] M. J. Kuehn and N. C. Kesty, "Bacterial outer membrane vesicles and the host–pathogen interaction," *Genes & Development*, vol. 19, no. 22, pp. 2645–2655, 2005.

[234] G. A. Kuldau, G. De Vos, J. Owen, G. McCaffrey, and P. Zambryski, "The virB operon of Agrobacterium tumefaciens pTiC58 encodes 11 open reading frames," *Molecular and General Genetics MGG*, vol. 221, no. 2, pp. 256–266, 1990.

[235] R. Kumar and B. Nanduri, "HPIDB-a unified resource for host-pathogen interactions," *BMC Bioinformatics*, vol. 11, no. 6, pp. 1–6, 2010.

[236] C. L. Kurz and J. J. Ewbank, "Caenorhabditis elegans for the study of host–pathogen interactions," *Trends in Microbiology*, vol. 8, no. 3, pp. 142–144, 2000.

[237] L. Lam and S. Suen, “Application of majority voting to pattern recognition: an analysis of its behavior and performance,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 27, no. 5, pp. 553–568, 1997.

[238] C. Landolt-Marticorena, K. A. Williams, C. M. Deber, and R. A. Reithmeier, “Non-random distribution of amino acids in the transmembrane segments of human type I single span membrane proteins,” *Journal of Molecular Biology*, vol. 229, no. 3, pp. 602–608, 1993.

[239] P. Larranaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armananzas, G. Santafé, A. Pérez *et al.*, “Machine learning in bioinformatics,” *Briefings in Bioinformatics*, vol. 7, no. 1, pp. 86–112, 2006.

[240] D. Lawson, P. Arensburger, P. Atkinson, N. J. Besansky, R. V. Bruggner, R. Butler, K. S. Campbell, G. K. Christophides, S. Christley, E. Dialynas *et al.*, “VectorBase: a home for invertebrate vectors of human pathogens,” *Nucleic Acids Research*, vol. 35, no. 1, pp. D503–D505, 2007.

[241] D. Lawson, P. Arensburger, P. Atkinson, N. J. Besansky, R. V. Bruggner, R. Butler, K. S. Campbell, G. K. Christophides, S. Christley, E. Dialynas *et al.*, “VectorBase: a data resource for invertebrate vector genomics,” *Nucleic Acids Research*, vol. 37, no. 1, pp. D583–D587, 2009.

[242] M. Lecuit, S. Dramsi, C. Gottardi, M. Fedor-Chaiken, B. Gumbiner, and P. Cossart, “A single amino acid in E-cadherin responsible for host specificity towards the human pathogen Listeria monocytogenes,” *The EMBO Journal*, vol. 18, no. 14, pp. 3956–3963, 1999.

[243] K.-Y. Lee and B.-J. Lee, “Structure, biology, and therapeutic application of toxin–antitoxin systems in pathogenic bacteria,” *Toxins*, vol. 8, no. 10, pp. 1–33, 2016.

[244] S.-A. Lee, C.-h. Chan, C.-H. Tsai, J.-M. Lai, F.-S. Wang, C.-Y. Kao, and C.-Y. F. Huang, “Ortholog-based protein-protein interaction prediction and its application to inter-species interactions,” *BMC Bioinformatics*, vol. 9, no. 12, pp. 1–9, 2008.

[245] A. J. Lees, “Advice for those trying to find a cure for the shaking palsy,” *The Lancet Neurology*, vol. 17, no. 1, pp. 27–28, 2018.

[246] P. G. Leiman, M. Basler, U. A. Ramagopal, J. B. Bonanno, J. M. Sauder, S. Pukatzki, S. K. Burley, S. C. Almo, and J. J. Mekalanos, “Type VI secretion apparatus and phage tail-associated protein complexes share a common evolutionary origin,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 11, pp. 4154–4159, 2009.

[247] N. Leo, J. Liu, I. Archbold, Y. Tang, and X. Zeng, “Ionic strength, surface charge, and packing density effects on the properties of peptide self-assembled monolayers,” *Langmuir*, vol. 33, no. 8, pp. 2050–2058, 2017.

[248] A. Lesk, *Introduction to Bioinformatics*. Oxford, United Kingdom:Oxford university press, 2019.

[249] V. Letchumanan, K.-G. Chan, and L.-H. Lee, “Vibrio parahaemolyticus: a review on the pathogenesis, prevalence, and advance molecular identification techniques,” *Frontiers in Microbiology*, vol. 5, no. 705, pp. 1–13, 2014.

[250] J. C. Lewthwaite, A. R. Coates, P. Tormay, M. Singh, P. Mascagni, S. Poole, M. Roberts, L. Sharp, and B. Henderson, “Mycobacterium tuberculosis Chaperonin 60.1 Is a More Potent Cytokine Stimulator than Chaperonin 60.2 (Hsp 65) and Contains a CD14-Binding Domain,” *Infection and Immunity*, vol. 69, no. 12, pp. 7349–7355, 2001.

[251] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang, “The yeast cell-cycle network is robustly designed,” *Proceedings of the National Academy of Sciences*, vol. 101, no. 14, pp. 4781–4786, 2004.

[252] G. Li, Y. Fan, Y. Lai, T. Han, Z. Li, P. Zhou, P. Pan, W. Wang, D. Hu, X. Liu *et al.*, “Coronavirus infections and immune responses,” *Journal of Medical Virology*, 2020.

[253] J. Li, Y. Yao, H. H. Xu, L. Hao, Z. Deng, K. Rajakumar, and H.-Y. Ou, “SecReT6: a web-based resource for Type VI secretion systems found in bacteria,” *Environmental Microbiology*, vol. 17, no. 7, pp. 2196–2202, 2015.

[254] N. Li, Y. Zhu, B. R. LaFrentz, J. P. Evenhuis, D. W. Hunnicutt, R. A. Conrad, P. Barbier, C. W. Gullstrand, J. E. Roets, J. L. Powers *et al.*, “The type IX secretion system is required for virulence of the fish pathogen flavobacterium columnare,” *Applied and Environmental Microbiology*, vol. 83, no. 23, pp. e01 769–17, 2017.

[255] N. J. Lightner, *Advances in Human Factors and Ergonomics in Healthcare and Medical Devices*. Walt Disney World, FL, USA: Springer, 2020.

[256] W. I. Lipkin, “The changing face of pathogen discovery and surveillance,” *Nature Reviews Microbiology*, vol. 11, no. 2, pp. 133–141, 2013.

[257] J. Liu, J. Zhu, L. Tang, W. Wen, S. Lv, and R. Yu, “Enhancement of vindoline and vinblastine production in suspension-cultured cells of Catharanthus roseus by artemisinic acid elicitation,” *World Journal of Microbiology and Biotechnology*, vol. 30, no. 1, pp. 175–180, 2014.

[258] T. Liu, X. Zheng, and J. Wang, “Prediction of protein structural class for low-similarity sequences using support vector machine and psi-blast profile,” *Biochimie*, vol. 92, no. 10, pp. 1330–1334, 2010.

[259] X. Liu, J. Wu, D. Zhang, K. Wang, X. Duan, Z. Meng, and X. Zhang, “Network pharmacology-based approach to investigate the mechanisms of Hedyotis diffusa willd. in the treatment of gastric cancer,” *Evidence-Based Complementary and Alternative Medicine*, vol. 17, 2018.

[260] D. H. P. Low, V. Frecer, A. Le Saux, G. A. Srinivasan, B. Ho, J. Chen, and J. L. Ding, “Molecular interfaces of the galactose-binding protein Tectonin domains in host-pathogen interaction,” *Journal of Biological Chemistry*, vol. 285, no. 13, pp. 9898–9907, 2010.

[261] M. Löwer and G. Schneider, “Prediction of Type III secretion signals in genomes of gram-negative bacteria,” *PloS One*, vol. 4, no. 6, p. e5917, 2009.

[262] Y. L. E. Lui, T. L. Tan, P. Timms, L. M. Hafner, K. H. Tan, and E. L. Tan, “Elucidating the host–pathogen interaction between human colorectal cells and invading Enterovirus 71 using transcriptomics profiling,” *FEBS Open Bio*, vol. 4, no. 1, pp. 426–431, 2014.

[263] B. Luscher, T. Fuchs, and C. L. Kilpatrick, “GABAA receptor trafficking-mediated plasticity of inhibitory synapses,” *Neuron*, vol. 70, no. 3, pp. 385–409, 2011.

[264] X. Lyu, C. Shen, Y. Fu, J. Xie, D. Jiang, G. Li, and J. Cheng, “A small secreted virulence-related protein is essential for the necrotrophic interactions of sclerotinia sclerotiorum with its host plants,” *PLoS Pathogens*, vol. 12, no. 2, p. e1005435, 2016.

[265] J. Ma, Z. Pan, J. Huang, M. Sun, C. Lu, and H. Yao, “The Hcp proteins fused with diverse extended-toxin domains represent a novel pattern of antibacterial effectors in type VI secretion systems,” *Virulence*, vol. 8, no. 7, pp. 1189–1202, 2017.

[266] R. Mariano and S. Wuchty, “Structure-based prediction of host–pathogen protein interactions,” *Current Opinion in Structural Biology*, vol. 44, pp. 119–124, 2017.

[267] H. M. Marriott, T. J. Mitchell, and D. H. Dockrell, “Pneumolysin: a double-edged sword during the host-pathogen interaction,” *Current Molecular Medicine*, vol. 8, no. 6, pp. 497–509, 2008.

[268] L. R. Matthews, P. Vaglio, J. Reboul, H. Ge, B. P. Davis, J. Garrels, S. Vincent, and M. Vidal, “Identification of potential interaction networks using sequence-based

searches for conserved protein-protein interactions or “interologs”,” *Genome Research*, vol. 11, no. 12, pp. 2120–2126, 2001.

[269] S. Mattoo, Y. M. Lee, and J. E. Dixon, “Interactions of bacterial effector proteins with host proteins,” *Current Opinion in Immunology*, vol. 19, no. 4, pp. 392–401, 2007.

[270] A. J. McCarthy and J. A. Lindsay, “Genetic variation in Staphylococcus aureus surface and immune evasion genes is lineage associated: implications for vaccine design and host-pathogen interactions,” *BMC Microbiology*, vol. 10, no. 1, pp. 1–15, 2010.

[271] J. E. McDermott, A. Corrigan, E. Peterson, C. Oehmen, G. Niemann, E. D. Cambronne, D. Sharp, J. N. Adkins, R. Samudrala, and F. Heffron, “Computational prediction of Type III and IV secreted effectors in gram-negative bacteria,” *Infection and Immunity*, vol. 79, no. 1, pp. 23–32, 2011.

[272] D. C. McShan, S. Rao, and I. Shah, “PathMiner: predicting metabolic pathways by heuristic search,” *Bioinformatics*, vol. 19, no. 13, pp. 1692–1698, 2003.

[273] F. Mech, A. Thywißen, R. Guthke, A. A. Brakhage, and M. T. Figge, “Automated image analysis of the host-pathogen interaction between phagocytes and Aspergillus fumigatus,” *PloS One*, vol. 6, no. 5, p. e19591, 2011.

[274] J. L. Meek, “Prediction of peptide retention times in high-pressure liquid chromatography on the basis of amino acid composition,” *Proceedings of the National Academy of Sciences*, vol. 77, no. 3, pp. 1632–1636, 1980.

[275] J.-L. Mege, “Dendritic cell subtypes: a new way to study host-pathogen interaction,” *Virulence*, vol. 7, no. 1, pp. 5–6, 2016.

[276] K. Megy, S. J. Emrich, D. Lawson, D. Campbell, E. Dialynas, D. S. Hughes, G. Koscielny, C. Louis, R. M. MacCallum, S. N. Redmond *et al.*, “VectorBase: improvements to a bioinformatics resource for invertebrate vector genomics,” *Nucleic Acids Research*, vol. 40, no. D1, pp. D729–D734, 2012.

[277] P. K. Meher, T. K. Sahu, V. Saini, and A. R. Rao, “Predicting antimicrobial peptides with improved accuracy by incorporating the compositional, physico-chemical and structural features into chou’s general pseaac,” *Scientific Reports*, vol. 7, no. 1, pp. 1–12, 2017.

[278] A. Mehra, A. Zahra, V. Thompson, N. Sirisaengtaksin, A. Wells, M. Porto, S. Köster, K. Penberthy, Y. Kubota, A. Dricot, D. Rogan, M. Vidal, D. Hill, A. Bean, and J. Philips, “Mycobacterium tuberculosis type VII secreted effector EsxH targets host ESCRT to impair trafficking,” *PLoS Pathogens*, vol. 9, no. 10, p. e1003734, 2013.

[279] D. Michie, D. J. Spiegelhalter, C. Taylor *et al.*, “Machine learning,” *Neural and Statistical Classification*, vol. 13, no. 1994, pp. 1–298, 1994.

[280] L. C. Miller, D. Fleming, A. Arbogast, D. O. Bayles, B. Guo, K. M. Lager, J. N. Henningson, S. N. Schlink, H.-C. Yang, K. S. Faaberg *et al.*, “Analysis of the swine tracheobronchial lymph node transcriptomic response to infection with a chinese highly pathogenic strain of porcine reproductive and respiratory syndrome virus,” *BMC Veterinary Research*, vol. 8, no. 1, pp. 1–8, 2012.

[281] A. K. Mishra, N. N. Driessen, B. J. Appelmelk, and G. S. Besra, “Lipoarabinomannan and related glycoconjugates: structure, biogenesis and role in Mycobacterium tuberculosis physiology and host–pathogen interaction,” *FEMS Microbiology Reviews*, vol. 35, no. 6, pp. 1126–1157, 2011.

[282] A. Mithani, G. M. Preston, and J. Hein, “Rahnuma: hypergraph-based tool for metabolic pathway prediction and network comparison,” *Bioinformatics*, vol. 25, no. 14, pp. 1831–1832, 2009.

[283] V. Molle and L. Kremer, “Division and cell envelope regulation by Ser/Thr phosphorylation: Mycobacterium shows the way,” *Molecular Microbiology*, vol. 75, no. 5, pp. 1064–1077, 2010.

[284] J. C. Moreno and T. J. Visser, “New phenotypes in thyroid dyshormonogenesis: hypothyroidism due to DUOX2 mutations,” in *Thyroid Gland Development and Function*. Karger Publishers, 2007, vol. 10, pp. 99–117.

[285] Y. Moriya, D. Shigemizu, M. Hattori, T. Tokimatsu, M. Kotera, S. Goto, and M. Kanehisa, “PathPred: an enzyme-catalyzed metabolic pathway prediction server,” *Nucleic Acids Research*, vol. 38, no. 2, pp. W138–W143, 2010.

[286] J. D. Mougous, M. E. Cuff, S. Raunser, A. Shen, M. Zhou, C. A. Gifford, A. L. Goodman, G. Joachimiak, C. L. Ordoñez, S. Lory *et al.*, “A virulence locus of Pseudomonas aeruginosa encodes a protein secretion apparatus,” *Science*, vol. 312, no. 5779, pp. 1526–1530, 2006.

[287] M.-H. Mucchielli-Giorgi, S. Hazout, and P. Tuffry, “PredAcc: prediction of solvent accessibility.” *Bioinformatics*, vol. 15, no. 2, pp. 176–177, 1999.

[288] R. E. Muir and M.-W. Tan, “Virulence of Leucobacter chromiireducens subsp. solipictus to Caenorhabditis elegans: characterization of a novel host-pathogen interaction,” *Applied and Environmental Microbiology*, vol. 74, no. 13, pp. 4185–4198, 2008.

[289] P. J. Murray and R. A. Young, “Stress and immunological recognition in host-pathogen interactions.” *Journal of Bacteriology*, vol. 174, no. 13, pp. 4193–4196, 1992.

[290] E. Mylonakis and A. Aballay, “Worms and flies as genetically tractable animal models to study host-pathogen interactions,” *Infection and Immunity*, vol. 73, no. 7, pp. 3833–3841, 2005.

[291] J. Naglik, A. Albrecht, O. Bader, and B. Hube, “Candida albicans proteinases and host/pathogen interactions,” *Cellular Microbiology*, vol. 6, no. 10, pp. 915–926, 2004.

[292] M. Nairz, A. Schroll, T. Sonnweber, and G. Weiss, “The struggle for iron–a metal at the host–pathogen interface,” *Cellular Microbiology*, vol. 12, no. 12, pp. 1691–1702, 2010.

[293] G. J. Nau, J. F. Richmond, A. Schlesinger, E. G. Jennings, E. S. Lander, and R. A. Young, “Human macrophage activation programs induced by bacterial pathogens,” *Proceedings of the National Academy of Sciences*, vol. 99, no. 3, pp. 1503–1508, 2002.

[294] V. Navratil, B. de Chassey, L. Meyniel, S. Delmotte, C. Gautier, P. André, V. Lotteau, and C. Rabourdin-Combe, “VirHostNet: a knowledge base for the management and the analysis of proteome-wide virus–host interaction networks,” *Nucleic Acids Research*, vol. 37, no. 1, pp. D661–D668, 2009.

[295] L. Nayak, R. K. De, and A. Datta, “Prediction of system states, robustness and stability of the human wnt signal transduction pathway using boolean logic,” in *BIOMAT 2015: International Symposium on Mathematical and Computational Biology*. World Scientific, Roorkee, UT, India, November 1-7, 2015, pp. 177–191.

[296] G. W. Newton, E. S. Schmidt, J. P. Lewis, R. Lawrence, and E. Conn, “Amygdalin toxicity studies in rats predict chronic cyanide poisoning in humans,” *Western Journal of Medicine*, vol. 134, no. 2, pp. 97–103, 1981.

[297] E. Nourani, F. Khunjush, and S. Durmuş, “Computational approaches for prediction of pathogen-host protein-protein interactions,” *Frontiers in Microbiology*, vol. 6, no. 94, pp. 1–10, 2015.

[298] J. S. Nowick and S. Insaf, “The propensities of amino acids to form parallel β-sheets,” *Journal of the American Chemical Society*, vol. 119, no. 45, pp. 10 903–10 908, 1997.

[299] M. Oh, T. Yamada, M. Hattori, S. Goto, and M. Kanehisa, "Systematic analysis of enzyme-catalyzed reaction patterns and prediction of microbial biodegradation pathways," *Journal of Chemical Information and Modeling*, vol. 47, no. 4, pp. 1702–1712, 2007.

[300] A. Ohara, F. Yamada, T. Fukuda, N. Suzuki, and K. Sumida, "Specific alteration of gene expression profile in rats by treatment with thyroid toxicants that inhibit thyroid hormone synthesis," *Journal of Applied Toxicology*, vol. 38, no. 12, pp. 1529–1537, 2018.

[301] J. E. Olsen, K. H. Hoegh-Andersen, J. Casadesús, J. Rosenkranzt, M. S. Chadfield, and L. E. Thomsen, "The role of flagella and chemotaxis genes in host pathogen interaction of the host adapted Salmonella enterica serovar Dublin compared to the broad host range serovar S. Typhimurium," *BMC Microbiology*, vol. 13, no. 1, pp. 1–11, 2013.

[302] G. Onder, G. Rezza, and S. Brusaferro, "Case-fatality rate and characteristics of patients dying in relation to COVID-19 in Italy," *Journal of the American Medical Association*, vol. 323, no. 18, pp. 1775–1776, 2020.

[303] D. W. Onstad, "Ecological database of the world's insect pathogens (edwip)," 1997. [Online]. Available: http://cricket.inhs.uiuc.edu/edwipweb/edwipabout.htm

[304] C. A. Orengo, A. E. Todd, and J. M. Thornton, "From protein structure to function," *Current Opinion in Structural Biology*, vol. 9, no. 3, pp. 374–382, 1999.

[305] E. Oswald, J.-P. Nougayrède, F. Taieb, and M. Sugai, "Bacterial toxins that modulate host cell-cycle progression," *Current Opinion in Microbiology*, vol. 8, no. 1, pp. 83–91, 2005.

[306] C. A. Ouzounis, "Rise and demise of bioinformatics? promise and progress," *PLoS Computational Biology*, vol. 8, no. 4, p. e1002487, 2012.

[307] C. N. Pace and J. M. Scholtz, "A helix propensity scale based on experimental studies of peptides and proteins," *Biophysical Journal*, vol. 75, no. 1, pp. 422–427, 1998.

[308] K. C. Pandey, N. Singh, S. Arastu-Kapur, M. Bogyo, and P. J. Rosenthal, "Falstatin, a cysteine protease inhibitor of Plasmodium falciparum, facilitates erythrocyte invasion," *PLoS Pathogen*, vol. 2, no. 11, p. e117, 2006.

[309] K. Pant, N. T. Devvret, A. Pandaya, and A. Thapliyal, "Mtb-HID: A unified database of host pathogen interaction for various Mycobacterium tuberculosis strains," *Current Sciences*, no. 2, pp. 292–297.

[310] C. Parikh, R. Subrahmanyam, and R. Ren, “Oncogenic NRAS, KRAS, and HRAS exhibit different leukemogenic potentials in mice,” *Cancer Research*, vol. 67, no. 15, pp. 7139–7146, 2007.

[311] K. Park, D. P. Williams, D. J. Naisbitt, N. R. Kitteringham, and M. Pirmohamed, “Investigation of toxic metabolites during drug development,” *Toxicology and Applied Pharmacology*, vol. 207, no. 2, pp. 425–434, 2005.

[312] J. S. Pearson, Y. Zhang, H. J. Newton, and E. L. Hartland, “Post-modern pathogens: surprising activities of translocated effectors from E. coli and Legionella,” *Current Opinion in Microbiology*, vol. 23, pp. 73–79, 2015.

[313] L. G. Pell, V. Kanelis, L. W. Donaldson, P. L. Howell, and A. R. Davidson, “The phage λ major tail protein structure reveals a common evolution for long-tailed phages and the type VI bacterial secretion system,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 11, pp. 4160–4165, 2009.

[314] P. R. Pereira, L. G. Fernandes, G. O. de Souza, S. A. Vasconcellos, M. B. Heinemann, E. C. Romero, and A. L. Nascimento, “Multifunctional and redundant roles of leptospira interrogans proteins in bacterial-adhesion and fibrin clotting inhibition,” *International Journal of Medical Microbiology*, vol. 307, no. 6, pp. 297–310, 2017.

[315] J. Pevsner, *Bioinformatics and Functional Genomics*. Hoboken, NJ, USA:John Wiley & Sons, 2015.

[316] B. E. Pickett, E. L. Sadat, Y. Zhang, J. M. Noronha, R. B. Squires, V. Hunt, M. Liu, S. Kumar, S. Zaremba, Z. Gu *et al.*, “ViPR: an open bioinformatics database and analysis resource for virology research,” *Nucleic Acids Research*, vol. 40, no. D1, pp. D593–D598, 2012.

[317] J. Pohlner, R. Halter, K. Beyreuther, and T. F. Meyer, “Gene structure and extracellular secretion of Neisseria gonorrhoeae IgA protease.” *Nature*, vol. 325, no. 6103, pp. 458–462, 1986.

[318] A. Poret and C. Guziolowski, “Therapeutic target discovery using boolean network attractors: improvements of kali,” *Royal Society Open Science*, vol. 5, no. 2, pp. 1–15, 2018.

[319] S. Pukatzki, A. T. Ma, D. Sturtevant, B. Krastins, D. Sarracino, W. C. Nelson, J. F. Heidelberg, and J. J. Mekalanos, “Identification of a conserved bacterial protein secretion system in Vibrio cholerae using the Dictyostelium host model system,” *Proceedings of the National Academy of Sciences*, vol. 103, no. 5, pp. 1528–1533, 2006.

[320] Y. Qi, O. Tastan, J. G. Carbonell, J. Klein-Seetharaman, and J. Weston, "Semi-supervised multi-task learning for predicting interactions between HIV-1 and human proteins," *Bioinformatics*, vol. 26, no. 18, pp. i645–i652, 2010.

[321] H. Rachman, M. Strong, T. Ulrichs, L. Grode, J. Schuchhardt, H. Mollenkopf, G. A. Kosmiadi, D. Eisenberg, and S. H. Kaufmann, "Unique transcriptome signature of Mycobacterium tuberculosis in pulmonary tuberculosis," *Infection and Immunity*, vol. 74, no. 2, pp. 1233–1242, 2006.

[322] A. Raghunathan, J. Reed, S. Shin, B. Palsson, and S. Daefler, "Constraint-based analysis of metabolic capacity of Salmonella typhimurium during host-pathogen interaction," *BMC Systems Biology*, vol. 3, no. 1, pp. 38–53, 2009.

[323] M. M. Rahman and D. Davis, "Addressing the class imbalance problem in medical datasets," *International Journal of Machine Learning and Computing*, vol. 3, no. 2, p. 224, 2013.

[324] S. A. Rahman, P. Advani, R. Schunk, R. Schrader, and D. Schomburg, "Metabolic pathway analysis web service (pathway hunter tool at cubic)," *Bioinformatics*, vol. 21, no. 7, pp. 1189–1193, 2005.

[325] S. A. Rahman, Y. Singh, S. Kohli, J. Ahmad, N. Z. Ehtesham, A. K. Tyagi, and S. E. Hasnain, "Comparative analyses of nonpathogenic, opportunistic, and totally pathogenic mycobacteria reveal genomic and biochemical variabilities and highlight the survival attributes of Mycobacterium tuberculosis," *mBio*, vol. 5, no. 6, 2014. [Online]. Available: https://mbio.asm.org/content/5/6/e02020-14

[326] S. Ranganathan, K. Nakai, and C. Schonbach, *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*. Camridge, MA, USA:Elsevier, 2018.

[327] E. A. Ranheim and T. J. Kipps, "Elevated expression of CD80 (B7/BB1) and other accessory molecules on synovial fluid mononuclear cell subsets in rheumatoid arthritis," *Arthritis & Rheumatism: Official Journal of the American College of Rheumatology*, vol. 37, no. 11, pp. 1637–1646, 1994.

[328] N. Rappoport and M. Linial, "Viral proteins acquired from a host converge to simplified domain architectures," *PLoS Computational Biology*, vol. 8, no. 2, p. e1002364, 2012.

[329] A. J. Reid and M. Berriman, "Genes involved in host–parasite interactions can be revealed by their correlated expression," *Nucleic Acids Research*, vol. 41, no. 3, pp. 1508–1518, 2012.

[330] M. Rescigno and P. Borrow, "The host-pathogen interaction: new themes from dendritic cell biology," *Cell*, vol. 106, no. 3, pp. 267–270, 2001.

[331] F. M. Richards, "Packing defects, cavities, volume fluctuations, and access to the interior of proteins. including some general comments on surface area and protein structure," *Carlsberg Research Communications*, vol. 44, no. 2, pp. 47–63, 1979.

[332] J. B. Robinson, M. V. Telepnev, I. V. Zudina, D. Bouyer, J. A. Montenieri, S. W. Bearden, K. L. Gage, S. L. Agar, S. M. Foltz, S. Chauhan *et al.*, "Evaluation of a Yersinia pestis mutant impaired in a thermoregulated type VI-like secretion system in flea, macrophage and murine models," *Microbial Pathogenesis*, vol. 47, no. 5, pp. 243–251, 2009.

[333] A. Rodríguez, D. Sosa, L. Torres, B. Molina, S. Frías, and L. Mendoza, "A boolean network model of the fa/brca pathway," *Bioinformatics*, vol. 28, no. 6, pp. 858–866, 2012.

[334] G. D. Rose, A. R. Geselowitz, G. J. Lesser, R. H. Lee, and M. H. Zehfus, "Hydrophobicity of amino acid residues in globular proteins," *Science*, vol. 229, no. 4716, pp. 834–838, 1985.

[335] F. Rousset, E. Garcia, and J. Banchereau, "Cytokine-induced proliferation and immunoglobulin production of human B lymphocytes triggered through their CD40 antigen." *The Journal of Experimental Medicine*, vol. 173, no. 3, pp. 705–710, 1991.

[336] S. Rupp and K. Sohn, *Host-pathogen interactions: Methods and Protocols*. Humana Press, 2009.

[337] A. B. Russell, S. B. Peterson, and J. D. Mougous, "Type VI secretion system effectors: poisons with a purpose," *Nature Reviews Microbiology*, vol. 12, no. 2, pp. 137–148, 2014.

[338] R. Samudrala, F. Heffron, and J. E. McDermott, "Accurate prediction of secreted substrates and identification of a conserved putative secretion signal for Type III secretion systems," *PLoS Pathogens*, vol. 5, no. 4, p. e1000375, 2009.

[339] T. G. Sana, K. A. Lugo, and D. M. Monack, "T6SS: The bacterial "fight club" in the host gut," *PLoS Pathogens*, vol. 13, no. 6, p. e1006325, 2017.

[340] P. Sansonetti, “Host–pathogen interactions: the seduction of molecular cross talk,” *Gut*, vol. 50, no. 3, pp. iii2–iii8, 2002.

[341] Y. G. Santin and E. Cascales, “Domestication of a housekeeping transglycosylase for assembly of a type VI secretion system,” *EMBO Reports*, vol. 18, no. 1, pp. 138–149, 2017.

[342] M. S. f. M. Santos, P. H. Abreu, P. J. García-Laencina, A. Simão, and A. Carvalho, “A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients,” *Journal of Biomedical Informatics*, vol. 58, pp. 49–59, 2015.

[343] C. M. Sassetti, D. H. Boyd, and E. J. Rubin, “Genes required for mycobacterial growth defined by high density mutagenesis,” *Molecular Microbiology*, vol. 48, no. 1, pp. 77–84, 2003.

[344] C. M. Sassetti and E. J. Rubin, “Genetic requirements for mycobacterial survival during infection,” *Proceedings of the National Academy of Sciences*, vol. 100, no. 22, pp. 12 989–12 994, 2003.

[345] C. A. Scanga, A. Bafica, C. G. Feng, A. W. Cheever, S. Hieny, and A. Sher, “MyD88-deficient mice display a profound loss in resistance to mycobacterium tuberculosis associated with partially impaired Th1 cytokine and nitric oxide synthase 2 expression,” *Infection and Immunity*, vol. 72, no. 4, pp. 2400–2404, 2004.

[346] V. Scaria, M. Hariharan, S. Maiti, B. Pillai, and S. K. Brahmachari, “Host-virus interaction: a new role for microRNAs,” *Retrovirology*, vol. 3, no. 1, pp. 68–76, 2006.

[347] V. Scaria, M. Hariharan, B. Pillai, S. Maiti, and S. K. Brahmachari, “Host–virus genome interactions: macro roles for microRNAs,” *Cellular Microbiology*, vol. 9, no. 12, pp. 2784–2794, 2007.

[348] K. Schaaf, S. R. Smith, A. Duverger, F. Wagner, F. Wolschendorf, A. O. Westfall, O. Kutsch, and J. Sun, “Mycobacterium tuberculosis exploits the PPM1A signaling pathway to block host macrophage apoptosis,” *Scientific Reports*, vol. 7, no. 42101, pp. 1–16, 2017.

[349] C. F. Schaefer, K. Anthony, S. Krupa, J. Buchoff, M. Day, T. Hannay, and K. H. Buetow, “PID: the pathway interaction database,” *Nucleic Acids Research*, vol. 37, no. 1, pp. D674–D679, 2009.

[350] D. Schnappinger, S. Ehrt, M. I. Voskuil, Y. Liu, J. A. Mangan, I. M. Monahan, G. Dolganov, B. Efron, P. D. Butcher, C. Nathan *et al.*, "Transcriptional adaptation of Mycobacterium tuberculosis within macrophages insights into the phagosomal environment," *The Journal of Experimental Medicine*, vol. 198, no. 5, pp. 693–704, 2003.

[351] R. Sen and R. K. De, "DeepT7: A deep neural network system for identification of Type VII effector proteins," Computational Biology and Chemistry, (under revision).

[352] R. Sen and R. K. De, "Boolean logic based network robustness analyzer (BNRA) and its application to a system of host-pathogen interactions," (under preparation).

[353] R. Sen, L. Nayak, and R. K. De, "PyPredT6: A python based prediction tool for identification of Type VI effector proteins," *Journal of Bioinformatics and Computational Biology*, vol. 17, no. 03, pp. 1 950 019 1–1 950 019 18, 2019.

[354] R. Sen, L. Nayak, and R. K. De, "A review on host–pathogen interactions: classification and prediction," *European Journal of Clinical Microbiology & Infectious Diseases*, vol. 35, no. 10, pp. 1581–1599, 2016.

[355] R. Sen, S. Tagore, and R. K. De, "Cluster quality based non-reductional (CQNR) oversampling technique and effector protein predictor based on 3D structure (EPP3D) of proteins," *Computers in Biology and Medicine*, vol. 112, no. 103374, pp. 1–13, 2019.

[356] R. Sen, S. Tagore, and R. K. De, "ASAPP: Architectural similarity-based automated pathway prediction system and its application in host-pathogen interactions," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 17, no. 2, pp. 506–515, 2020.

[357] O. M. Sessions, N. J. Barrows, J. A. Souza-Neto, T. J. Robinson, C. L. Hershey, M. A. Rodgers, J. L. Ramirez, G. Dimopoulos, P. L. Yang, J. L. Pearson *et al.*, "Discovery of insect and human dengue virus host factors," *Nature*, vol. 458, no. 7241, pp. 1047–1050, 2009.

[358] A. Shahmoradi and C. O. Wilke, "Dissecting the roles of local packing density and longer-range effects in protein sequence evolution," *Proteins: Structure, Function, and Bioinformatics*, vol. 84, no. 6, pp. 841–854, 2016.

[359] O. P. Sharma, A. Jadhav, A. Hussain, and M. S. Kumar, "VPDB: Viral protein structural database," *Bioinformation*, vol. 6, no. 8, pp. 324–326, 2011.

[360] L. Shi, Q. Jiang, Y. Bushkin, S. Subbian, and S. Tyagi, “Biphasic dynamics of macrophage immunometabolism during Mycobacterium tuberculosis infection,” *MBio*, vol. 10, no. 2, pp. 1–19, 2019.

[361] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, “Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks,” *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.

[362] B. A. Shoemaker and A. R. Panchenko, “Deciphering protein–protein interactions: Computational methods to predict protein and domain interaction partners,” *PLoS Computational Biology*, vol. 3, no. 4, pp. 595–601, 2007.

[363] A. Y. Sim, P. Minary, and M. Levitt, “Modeling nucleic acids,” *Current Opinion in Structural Biology*, vol. 22, no. 3, pp. 273–278, 2012.

[364] R. Simeone, D. Bottai, and R. Brosch, “ESX”/type VII secretion systems and their role in host-pathogen interaction,” *Current Opinion in Microbiology*, vol. 12, no. 1, pp. 4–10, 2009.

[365] N. Singh, V. Bhatia, S. Singh, and S. Bhatnagar, “MorCVD: A unified database for host-pathogen protein-protein interactions of cardiovascular diseases related to microbes,” *Scientific Reports*, vol. 9, no. 1, pp. 1–10, 2019.

[366] S. B. Singh, A. S. Davis, G. A. Taylor, and V. Deretic, “Human IRGM induces autophagy to eliminate intracellular mycobacteria,” *Science*, vol. 313, no. 5792, pp. 1438–1441, 2006.

[367] V. Sintchenko, B. Gallego, G. Chung, and E. Coiera, “Towards bioinformatics assisted infectious disease control,” *BMC Bioinformatics*, vol. 10, no. 2, pp. 1–9, 2009.

[368] J. Sklansky, “Finding the convex hull of a simple polygon,” *Pattern Recognition Letters*, vol. 1, no. 2, pp. 79–83, 1982.

[369] D. N. Slenter, M. Kutmon, K. Hanspers, A. Riutta, J. Windsor, N. Nunes, J. Mélius, E. Cirillo, S. L. Coort, D. Digles *et al.*, “WikiPathways: a multifaceted pathway database bridging metabolomics to other omics research,” *Nucleic Acids Research*, vol. 46, no. D1, pp. D661–D667, 2018.

[370] J. S. Smith, *Patenting the sun: Polio and the Salk vaccine*. New York, NY, USA: William Morrow and Company, 1990.

[371] D. Smoot, H. Mobley, G. Chippendale, J. Lewison, and J. Resau, "Helicobacter pylori urease activity is toxic to human gastric epithelial cells." *Infection and Immunity*, vol. 58, no. 6, pp. 1992–1994, 1990.

[372] P. Sneath, "Relations between chemical structure and biological activity in peptides," *Journal of Theoretical Biology*, vol. 12, no. 2, pp. 157–195, 1966.

[373] K. H. Sohn, R. K. Hughes, S. J. Piquerez, J. D. Jones, and M. J. Banfield, "Distinct regions of the Pseudomonas syringae coiled-coil effector AvrRps4 are required for activation of immunity," *Proceedings of the National Academy of Sciences*, vol. 109, no. 40, pp. 16 371–16 376, 2012.

[374] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[375] H. L. Spiewak, S. Shastri, L. Zhang, S. Schwager, L. Eberl, A. C. Vergunst, and M. S. Thomas, "Burkholderia cenocepacia utilizes a Type VI secretion system for bacterial competition," *Microbiology Open*, p. e774, 2019.

[376] B. Squires, C. Macken, A. Garcia-Sastre, S. Godbole, J. Noronha, V. Hunt, R. Chang, C. N. Larsen, E. Klem, K. Biersack *et al.*, "BioHealthBase: informatics support in the elucidation of influenza virus host–pathogen interactions and virulence," *Nucleic Acids Research*, vol. 36, no. 1, pp. D497–D503, 2008.

[377] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[378] S. A. Stanley, S. Raghavan, W. W. Hwang, and J. S. Cox, "Acute infection and macrophage subversion by Mycobacterium tuberculosis require a specialized secretion system," *Proceedings of the National Academy of Sciences*, vol. 100, no. 22, pp. 13 001–13 006, 2003.

[379] D. R. Stockwell and A. T. Peterson, "Effects of sample size on accuracy of species distribution models," *Ecological Modelling*, vol. 148, no. 1, pp. 1–13, 2002.

[380] M. Stout, J. Bacardit, J. D. Hirst, and N. Krasnogor, "Prediction of recursive convex hull class assignments for protein residues," *Bioinformatics*, vol. 24, no. 7, pp. 916–923, 2008.

[381] G. Suarez, J. C. Sierra, J. Sha, S. Wang, T. E. Erova, A. A. Fadl, S. M. Foltz, A. J. Horneman, and A. K. Chopra, “Molecular characterization of a functional type VI secretion system from a clinical isolate of Aeromonas hydrophila,” *Microbial Pathogenesis*, vol. 44, no. 4, pp. 344–361, 2008.

[382] S. Sugimoto, T. Iwamoto, K. Takada, K.-i. Okuda, A. Tajima, T. Iwase, and Y. Mizunoe, “Staphylococcus epidermidis Esp degrades specific proteins associated with Staphylococcus aureus biofilm formation and host-pathogen interaction,” *Journal of Bacteriology*, vol. 195, no. 8, pp. 1645–1655, 2013.

[383] M. Sugiyama and M. Kawanabe, *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. Cambridge, MA, USA:MIT press, 2012.

[384] S. Tagore and R. K. De, “SAGPAR: Structural grammar-based automated pathway reconstruction,” *Interdisciplinary Sciences: Computational Life Sciences*, vol. 4, no. 2, pp. 116–127, 2012.

[385] A. M. Talaat, R. Lyons, S. T. Howard, and S. A. Johnston, “The temporal expression profile of Mycobacterium tuberculosis infection in mice,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 13, pp. 4602–4607, 2004.

[386] A. J. Tallón-Ballesteros and J. C. Riquelme, “Data mining methods applied to a digital forensics task for supervised machine learning,” in *Computational Intelligence in Digital Forensics: Forensic Investigation and Applications*. Springer, 2014, pp. 413–428.

[387] F. Tang and M. H. Saier Jr, “Transport proteins promoting Escherichia coli pathogenesis,” *Microbial Pathogenesis*, vol. 71, pp. 41–55, 2014.

[388] O. Tastan, Y. Qi, J. G. Carbonell, and J. Klein-Seetharaman, “Prediction of interactions between HIV-1 and human proteins by information integration,” in *Pacific Symposium on Biocomputing*. NIH Public Access, Kohala Coast, HI, USA, January 5-9, 2009, pp. 516–527.

[389] C. Tato and C. Hunter, “Host-pathogen interactions: subversion and utilization of the NF-κB pathway during infection,” *Infection and Immunity*, vol. 70, no. 7, pp. 3311–3317, 2002.

[390] S. D. Tekir, T. Çakır, E. Ardıç, A. S. Sayılırbaş, G. Konuk, M. Konuk, H. Sarıyer, A. Uğurlu, İ. Karadeniz, A. Özgür *et al.*, “PHISTO: pathogen–host interaction search tool,” *Bioinformatics*, vol. 29, no. 10, pp. 1357–1358, 2013.

[391] A. Tharwat, “Classification assessment methods,” *Applied Computing and Informatics*, pp. 1–13, 2018.

[392] T. Thieu, S. Joshi, S. Warren, and D. Korkin, “Literature mining of host–pathogen interactions: comparing feature-based supervised learning and language-based approaches,” *Bioinformatics*, vol. 28, no. 6, pp. 867–875, 2012.

[393] R. Thomas, “Boolean formalization of genetic control circuits,” *Journal of Theoretical Biology*, vol. 42, no. 3, pp. 563–585, 1973.

[394] A. Tidhar, Y. Flashner, S. Cohen, Y. Levi, A. Zauberman, D. Gur, M. Aftalion, E. Elhanany, A. Zvi, A. Shafferman *et al.*, “The NlpD lipoprotein is a novel Yersinia pestis virulence factor essential for the development of plague,” *PloS One*, vol. 4, no. 9, p. e7023, 2009.

[395] G. S. Tillotson and J. Tillotson, “Bacterial Secreted Proteins: Secretory Mechanisms and Role in Pathogenesis,” *Expert Review of Anti-infective Therapy*, vol. 7, no. 6, pp. 691–693, 2009.

[396] D. M. Tobin, R. C. May, and R. T. Wheeler, “Zebrafish: a see-through host and a fluorescent toolbox to probe host-pathogen interaction,” *PLoS Pathogens*, vol. 8, no. 1–3, 2012.

[397] K. Todar, *Mechanisms of Bacterial Pathogenicity*, http://textbookofbacteriology.net/pathogenesis.html.

[398] J. B. Torrelles and L. S. Schlesinger, “Diversity in Mycobacterium tuberculosis mannosylated cell wall determinants impacts adaptation to the host,” *Tuberculosis*, vol. 90, no. 2, pp. 84–93, 2010.

[399] L. Van Loon, “The induction of pathogenesis-related proteins by pathogens and specific chemicals,” *Netherlands Journal of Plant Pathology*, vol. 89, no. 6, pp. 265–279, 1983.

[400] A. Varshavsky, “The ubiquitin system,” *Trends in Biochemical Sciences*, vol. 22, no. 10, pp. 383–387, 1997.

[401] W. N. Venables and B. D. Ripley, “Tree-based methods,” in *Modern Applied Statistics with S*. Springer, 2002, pp. 251–269.

[402] I. Vergne, S. Singh, E. Roberts, G. Kyei, S. Master, J. Harris, S. d. Haro, J. Naylor, A. Davis, M. Delgado *et al.*, “Autophagy in immune defense against Mycobacterium tuberculosis,” *Autophagy*, vol. 2, no. 3, pp. 175–178, 2006.

[403] A. Via, B. Uyar, C. Brun, and A. Zanzoni, "How pathogens use linear motifs to perturb host cell networks," *Trends in Biochemical Sciences*, vol. 40, no. 1, pp. 36–48, 2015.

[404] N. Vodovar, C. Acosta, B. Lemaitre, and F. Boccard, "Drosophila: a polyvalent model to decipher host–pathogen interactions," *Trends in Microbiology*, vol. 12, no. 5, pp. 235–242, 2004.

[405] P. Vora, B. Oza *et al.*, "A survey on k-mean clustering and particle swarm optimization," *International Journal of Science and Modern Engineering*, vol. 1, no. 3, pp. 1–14, 2013.

[406] J. Wang, J. Li, B. Yang, R. Xie, T. T. Marquez-Lago, A. Leier, M. Hayashida, T. Akutsu, Y. Zhang, K.-C. Chou *et al.*, "Bastion3: a two-layer ensemble predictor of Type III secreted effectors," *Bioinformatics*, vol. 35, no. 12, pp. 2017–2028, 2018.

[407] J. Wang, J. Li, B. Yang, R. Xie, T. T. Marquez-Lago, A. Leier, M. Hayashida, T. Akutsu, Y. Zhang, K.-C. Chou *et al.*, "Bastion3: a two-layer ensemble predictor of Type III secreted effectors," *Bioinformatics*, vol. 35, no. 12, pp. 2017–2028, 2019.

[408] J. Wang, B. Yang, Y. An, T. Marquez-Lago, A. Leier, J. Wilksch, Q. Hong, Y. Zhang, M. Hayashida, T. Akutsu *et al.*, "Systematic analysis and prediction of Type IV secreted effector proteins by machine learning approaches," *Briefings in Bioinformatics*, vol. 20, no. 3, pp. 931–951, 2017.

[409] J. Wang, B. Yang, A. Leier, T. T. Marquez-Lago, M. Hayashida, A. Rocker, Y. Zhang, T. Akutsu, K.-C. Chou, R. A. Strugnell *et al.*, "Bastion6: a bioinformatics approach for accurate prediction of type VI secreted effectors," *Bioinformatics*, vol. 34, no. 15, pp. 2546–2555, 2018.

[410] N. Wang, Y. Wu, M. Pang, J. Liu, C. Lu, and Y. Liu, "Protective efficacy of recombinant hemolysin co-regulated protein (Hcp) of aeromonas hydrophila in common carp (Cyprinus carpio)," *Fish & Shellfish Immunology*, vol. 46, no. 2, pp. 297–304, 2015.

[411] W. Wang, S. Nag, X. Zhang, M.-H. Wang, H. Wang, J. Zhou, and R. Zhang, "Ribosomal proteins and human diseases: pathogenesis, molecular mechanisms, and therapeutic implications," *Medicinal Research Reviews*, vol. 35, no. 2, pp. 225–285, 2015.

[412] Y. Wang, Q. Zhang, M.-a. Sun, and D. Guo, "High-accuracy prediction of bacterial type III secreted effectors based on position-specific amino acid composition profiles," *Bioinformatics*, vol. 27, no. 6, pp. 777–784, 2011.

[413] Y. Wang, Y. Guo, X. Pu, and M. Li, “Effective prediction of bacterial Type IV secreted effectors by combined features of both c-termini and n-termini,” *Journal of Computer-aided Molecular Design*, vol. 31, no. 11, pp. 1029–1038, 2017.

[414] Z. Wang and J. Moult, “SNPs, protein structure, and disease,” *Human Mutation*, vol. 17, no. 4, pp. 263–270, 2001.

[415] C. Wanjek, “Systems biology as defined by nih: an intellectual resource for integrative biology,” *The NIH Catalyst*, vol. 19, no. 6, pp. 1–12, 2011.

[416] B. Warne, C. P. Harkins, S. R. Harris, A. Vatsiou, N. Stanley-Wall, J. Parkhill, S. J. Peacock, T. Palmer, and M. T. Holden, “The ess/type vii secretion system of staphylococcus aureus shows unexpected genetic diversity,” *BMC Genomics*, vol. 17, no. 1, pp. 1–13, 2016.

[417] M. S. Waterman, *Introduction to Computational Biology: Maps, Sequences and Genomes*. Boca Raton, FL, USA:Chapman and Hall/CRC, 2018.

[418] A. R. Wattam, D. Abraham, O. Dalay, T. L. Disz, T. Driscoll, J. L. Gabbard, J. J. Gillespie, R. Gough, D. Hix, R. Kenyon *et al.*, “PATRIC, the bacterial bioinformatics database and analysis resource,” *Nucleic Acids Research*, p. gkt1099, 2013.

[419] M. P. Weekes, P. Tomasec, E. L. Huttlin, C. A. Fielding, D. Nusinow, R. J. Stanton, E. C. Wang, R. Aicheler, I. Murrell, G. W. Wilkinson *et al.*, “Quantitative temporal viromics: an approach to investigate host-pathogen interaction,” *Cell*, vol. 157, no. 6, pp. 1460–1472, 2014.

[420] B. A. Weigele, R. C. Orchard, A. Jimenez, G. W. Cox, and N. M. Alto, “A systematic exploration of the interactions between bacterial effector proteins and host cell membranes,” *Nature Communications*, vol. 8, no. 1, pp. 1–14, 2017.

[421] R. Welch, E. Dellinger, B. Minshew, and S. Falkow, “Haemolysin contributes to virulence of extra-intestinal E. coli infections.” *Nature*, vol. 294, no. 5842, pp. 665–667, 1981.

[422] D. Whitford, *Proteins: Structure and Function*. Chichester, WS, England: John Wiley & Sons, 2013.

[423] R. Winnenburg, T. K. Baldwin, M. Urban, C. Rawlings, J. Köhler, and K. E. Hammond-Kosack, “PHI-base: a new database for pathogen host interactions,” *Nucleic Acids Research*, vol. 34, no. 1, pp. D459–D464, 2006.

[424] R. Winnenburg, M. Urban, A. Beacham, T. K. Baldwin, S. Holland, M. Lindeberg, H. Hansen, C. Rawlings, K. E. Hammond-Kosack, and J. Köhler, "PHI-base update: additions to the pathogen–host interaction database," *Nucleic Acids Research*, vol. 36, no. 1, pp. D572–D576, 2008.

[425] K.-C. Wong, *Computational Biology and Bioinformatics: Gene regulation.* Boca Raton, FL, USA:CRC Press, 2016.

[426] S. Wuchty, "Computational prediction of host-parasite protein interactions between P. falciparum and H. sapiens," *PLoS One*, vol. 6, no. 11, p. e26960, 2011.

[427] O. Wurtzel, N. Sesto, J. R. Mellin, I. Karunker, S. Edelheit, C. Bécavin, C. Archambaud, P. Cossart, and R. Sorek, "Comparative transcriptomics of pathogenic and non-pathogenic listeria species," *Molecular Systems Biology*, vol. 8, no. 1, pp. 583–596, 2012.

[428] B. Xayarath, H. Marquis, G. C. Port, and N. E. Freitag, "Listeria monocytogenes CtaP is a multifunctional cysteine transport-associated protein required for bacterial pathogenesis," *Molecular Microbiology*, vol. 74, no. 4, pp. 956–973, 2009.

[429] Z. Xiang, Y. Tian, Y. He *et al.*, "PHIDIAS: a pathogen-host interaction data integration and analysis system," *Genome Biology*, vol. 8, no. 7, pp. 1–15, 2007.

[430] Y. Xie, W. Hou, X. Song, Y. Yu, J. Huang, X. Sun, R. Kang, and D. Tang, "Ferroptosis: process and function," *Cell Death & Differentiation*, vol. 23, no. 3, pp. 369–379, 2016.

[431] J. Xiong, *Essential Bioinformatics.* New York, NY, USA:Cambridge University Press, 2006.

[432] Y. Xiong, Q. Wang, J. Yang, X. Zhu, and D. Wei, "PredT4SE-Stack: prediction of bacterial Type IV secreted effectors from protein sequences using a stacked ensemble method," *Frontiers in Microbiology*, vol. 9, no. 2571, pp. 1–9, 2018.

[433] L. Xue, B. Tang, W. Chen, and J. Luo, "DeepT3: deep convolutional neural networks accurately identify gram-negative bacterial Type III secreted effectors using the n-terminal sequence," *Bioinformatics*, vol. 35, no. 12, pp. 2051–2057, 2019.

[434] A. Yachie-Kinoshita, K. Onishi, J. Ostblom, M. A. Langley, E. Posfai, J. Rossant, and P. W. Zandstra, "Modeling signaling-dependent pluripotency with boolean logic to predict cell fate transitions," *Molecular Systems Biology*, vol. 14, no. 1, p. e7952, 2018.

[435] C. Yang, W. Jie, Y. Yanlong, G. Xuefeng, T. Aihua, G. Yong, L. Zheng, Z. Youjie, Z. Haiying, Q. Xue *et al.*, "Genome-wide association study identifies TNFSF13 as a susceptibility gene for IgA in a south chinese population in smokers," *Immunogenetics*, vol. 64, no. 10, pp. 747–753, 2012.

[436] J. Yang, L. Chen, L. Sun, J. Yu, and Q. Jin, "VFDB 2008 release: an enhanced web-based resource for comparative pathogenomics," *Nucleic Acids Research*, vol. 36, no. 1, pp. D539–D542, 2008.

[437] J. Yang, Y. Zheng, X. Gou, K. Pu, Z. Chen, Q. Guo, R. Ji, H. Wang, Y. Wang, and Y. Zhou, "Prevalence of comorbidities in the novel wuhan coronavirus (COVID-19) infection: a systematic review and meta-analysis," *International Journal of Infectious Diseases*, vol. 94, pp. 91–95, 2020.

[438] X. Yang, J. Tong, L. Guo, Z. Qian, Q. Chen, R. Qi, and Y. Qiu, "Bundling potent natural toxin cantharidin within platinum (IV) prodrugs for liposome drug delivery and effective malignant neuroblastoma treatment," *Nanomedicine: Nanotechnology, Biology and Medicine*, vol. 13, no. 1, pp. 287–296, 2017.

[439] X. Yang, Y. Guo, J. Luo, X. Pu, and M. Li, "Effective identification of Gram-negative bacterial type III secreted effectors using position-specific residue conservation profiles," *PloS One*, vol. 8, no. 12, p. e84439, 2013.

[440] Y. Yang, J. Zhao, R. L. Morgan, W. Ma, and T. Jiang, "Computational prediction of Type III secreted proteins from gram-negative bacteria," *BMC Bioinformatics*, vol. 11, no. 1, pp. 1–10, 2010.

[441] C. C. C. Yap, Z. M. Lasiecka, S. Caplan, and B. Winckler, "Alterations of EHD1/EHD4 protein levels interfere with L1/NgCAM endocytosis in neurons and disrupt axonal targeting," *Journal of Neuroscience*, vol. 30, no. 19, pp. 6646–6657, 2010.

[442] J. Yerushalmy, "Statistical problems in assessing methods of medical diagnosis, with special reference to X-ray techniques," *Public Health Reports (1896-1970)*, pp. 1432–1449, 1947.

[443] P. Yildirim and D. Birant, "The relative performance of deep learning and ensemble learning for textile object classification," in *2018 3rd International Conference on Computer Science and Engineering (UBMK)*. IEEE, Sarajevo, Bosnia-Herzegovina, September 23-28, 2018, pp. 22–26.

[444] J. Zahiri, O. Yaghoubi, M. Mohammad-Noori, R. Ebrahimpour, and A. Masoudi-Nejad, “PPIevo: Protein–protein interaction prediction from PSSM based evolutionary information,” *Genomics*, vol. 102, no. 4, pp. 237–242, 2013.

[445] A. Zalguizuri, G. Caetano-Anollés, and V. C. Lepek, “Phylogenetic profiling, an untapped resource for the prediction of secreted proteins and its complementation with sequence-based classifiers in bacterial type III, IV and VI secretion systems,” *Briefings in Bioinformatics*, vol. 20, no. 4, pp. 1395–1402, 2018.

[446] E. L. Zechner, S. Lang, and J. F. Schildbach, “Assembly and mechanisms of bacterial Type IV secretion machines,” *Philosophical Transactions of the Royal Society B*, vol. 367, no. 1592, pp. 1073–1087, 2012.

[447] M. H. Zehfus and G. D. Rose, “Compact units in proteins,” *Biochemistry*, vol. 25, no. 19, pp. 5759–5765, 1986.

[448] M. R. Zelle, “Genetic constitutions of host and pathogen in mouse typhoid,” *Journal of Infectious Diseases*, vol. 71, no. 2, pp. 131–152, 1942.

[449] J. Zhang, H. Li, Y. Zhang, C. Zhao, Y. Zhu, and M. Han, “Uncovering the pharmacological mechanism of stemazole in the treatment of neurodegenerative diseases based on a network pharmacology approach,” *International Journal of Molecular Sciences*, vol. 21, no. 2, pp. 1–16, 2020.

[450] S. Zhang and X. Duan, “Prediction of protein subcellular localization with oversampling approach and chou’s general PseAAC,” *Journal of Theoretical Biology*, vol. 437, pp. 239–250, 2018.

[451] Y.-Q. Zhang and J. C. Rajapakse, *Machine Learning in Bioinformatics*. Hoboken, NJ, USA:Wiley Online Library, 2009, vol. 4.

[452] Y.-D. Zhang, Y. Zhang, P. Phillips, Z. Dong, and S. Wang, “Synthetic minority oversampling technique and fractal dimension for identifying multiple sclerosis,” *Fractals*, vol. 25, no. 04, p. 1740010, 2017.

[453] Y.-D. Zhang, G. Zhao, J. Sun, X. Wu, Z.-H. Wang, H.-M. Liu, V. V. Govindaraj, T. Zhan, and J. Li, “Smart pathological brain detection by synthetic minority oversampling technique, extreme learning machine, and jaya algorithm,” *Multimedia Tools and Applications*, vol. 77, no. 17, pp. 1–20, 2017.

[454] C. Zhou, J. Smith, M. Lam, A. Zemla, M. D. Dyer, and T. Slezak, “MvirDB-a microbial database of protein toxins, virulence factors and antibiotic resistance genes for

bio-defence applications," *Nucleic Acids Research*, vol. 35, no. 1, pp. D391–D394, 2007.

[455] H. Zhou, M. Xu, Q. Huang, A. T. Gates, X. D. Zhang, J. C. Castle, E. Stec, M. Ferrer, B. Strulovici, D. J. Hazuda *et al.*, "Genome-scale RNAi screen for host factors required for HIV replication," *Cell Host & Microbe*, vol. 4, no. 5, pp. 495–504, 2008.

[456] L. Zou, C. Nan, and F. Hu, "Accurate prediction of bacterial Type IV secreted effectors using amino acid composition and PSSM profiles," *Bioinformatics*, vol. 29, no. 24, pp. 3135–3142, 2013.

[457] A. Zychlinsky and P. J. Sansonetti, "Apoptosis as a proinflammatory event: what can we learn from bacteria-induced cell death?" *Trends in Microbiology*, vol. 5, no. 5, pp. 201–204, 1997.