Python Framework for Qualitative Modelling of Biological Regulatory Networks

By

Talha Junaid

Fall 2017-MS CSE 00000205488

Supervisor

Dr. Muhammad Tariq Saeed

A thesis submitted in partial fulfillment of
The requirement of the degree of

Master of Computational Science & Engineering

Department of Computational Sciences
Research Center for Modelling and Simulation (RCMS)
National University of Sciences and Technology (NUST)
Islamabad, Pakistan

Aug 2021

Declaration

I, Talha Junaid declare that this thesis titled "Python Framework for Qualitative Modelling of Biological Regulatory Networks" and the work presented in it are my own and has been generated by me as a result of my own original research. I confirm that:

1. This work was done wholly or mainly while in the candidature for a Master of Science degree at NUST
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at NUST or any other institution, this has been clearly stated
3. Where I have consulted the published work of others, this is always clearly attributed. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work
4. I have acknowledged all main sources of help
5. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself

---

Talha Junaid,
00000205488

Copyright Notice

This thesis is dedicated to *my beloved parents* and *teachers*

Abstract

The modelling and analysis of biological networks is a key challenge in understanding the functioning of complex cellular systems. Qualitative modelling framework introduced by René Thomas is a well-established method to study the dynamical behavior of biological networks. Significant contributions have been made towards the application of the qualitative modelling framework. However, the availability of the framework as an API and open-source implementation remains a challenge to date. In this work, we have developed an open source python package that can be used by application developers in the area of systems biology and bioinformatics towards analysis of qualitative biological regulatory networks. We demonstrate that our implementation can be used to create qualitative models and perform key computations, such as cycles, deadlock states and generation of dynamic state graphs. We report the processing time for various models from the literature. Moreover, a frontend application is developed as a part of the API is provided to facilitate development of future applications. In future, we aim to improve the scalability of our API for complex networks by employing model reduction and parallel processing techniques.

Keywords: *systems biology, biological regulatory networks, modelling, René Thomas, smbionet, genoteche, ginsim*

Acknowledgements

In the name of Allah, the Most Gracious and the Most Merciful, all praises to Allah for the strengths and His blessings in completing this thesis. First and foremost, I would like to express my greatest gratitude to my supervisor Dr. Muhammad Tariq Saeed, who has motivated, guided and helped me a lot throughout this research work and believed in me. His invaluable help of constructive comments and suggestions throughout the research and thesis work have contributed to the success of this study.

I'd like to extend my thanks to NUST and I treasure the research environment provided by RCMS. I am thankful to my Guidance and Examination Committee (GEC) members; Dr. Mian Ilyas Ahmad, Dr. Ishrat Jabeen (RCMS) and Dr. Amjad Ali (ASAB) who provided me with constant assistance throughout my research phase.

I would like to express my appreciation to Principal RCMS, Head of Department of Computational Sciences, all my teachers, all the technicians, attendants, exam branch, student affairs and admin staff for their co-operations. Sincere thanks to all my friends and class fellows for their kindness and moral support during my study. Thanks for the friendship and memories.

Contents

List of Figures

Listings

List of Tables

List of Abbreviations and Symbols

Abbreviations

BRN	Biological Regulatory Network
GRN	Gene Regulatory Network
WWW	World Wide Web
GNA	Genetic Network Analyzer
PL	Piecewise Linear
GIS	Geographic Information System
GSPS	General Systems Problem Solver
FIR	Fuzzy Inductive Reasoning
AI	Artificial Intelligence
SPNs	Stochastic Petri Nets
LSI	Laser Spot Imaging
PL	Photo Luminescence
NC	Nano Crystal
GPD	Generalized Pareto Distribution
2AR	2-Adrenergic Receptor

SDE	Stochastic Differential Equation
GINsim	Gene Interaction Network simulation
CTL	Computational Tree Logic
IDE	Integrated Development Environment

Chapter 1

Introduction

1.1 Biological Regulatory Networks

The relationship between biological entities is represented by Biological Regulator Networks (BRNs). Conventionally, in an initial step, it depends on the description of a graph of regularity, and each node in this graph represents a regulatory parameter. The regulator may be protein, RNA, DNA, and combination of both. Plenty of complex operations can be involved as their regulations, but the regulations' complexity is commonly clarified by performing two operations that are inhibition and activation which are denoted by positive and negative arcs in a graph. These arcs show the relation between the target and source nodes. Each of these nodes is modelled as a unique feature and must have a finite possible value. These values are normally boolean (only two values), i.e. 0 or 1 that denote protein presence (activity) or absence (inactivity). The interaction can be direct or indirect. For each node, the boolean function is defined. The purpose of this function is to detect the changes of its values as the values of the regulatory changes. In a regulatory graph, the recommended order of the values of all the nodes is required. According to the logical function, the state of the network can be upgraded, and logical functions are used on each node by activating a transition toward a successor state. In synchronous updating mode, all the nodes are updated at the same time till final state, also known as Unique Successor State (USS). So, the obtained dynamical performance is significantly determinate thoroughly. On the other hand, in an asynchronous updating mode, at a time only one node is updated. Several extensions and variants of the updating modes are already defined, for example allocating

probabilities for updates' node or allocating predetermined priorities etc.

René Thomas justified the Boolean modelling approach as a discretization of the continuous differential equation system [1], it opposed to the classical analysis in terms of continuous differential equations [2]. Snoussi and Thomas proved that a discrete approach can find all steady states in a given BRN [3]. In a later study, Thomas and Kaufman [4] produced the discrete formalism yields a qualitative alternative of the differential equations with a few possibilities of values for the parameters. An interesting study of Cinquin [5] based on the formal method of René Thomas showed the basis to develop a positive and negative regulation and its analysis on BRN.

1.2 Modelling of Biological Regulatory Networks

In BRNs, a group of molecular entities may be observed. These entities regulate and interact with each other. In the cell, the biological entities are represented by the nodes in the network such as proteins, genes, and their products. The edges are used to represent the interactions. Regulations (inhibition or activation) are shown by interactions. As a unique feature, each node is modelled, having a possible value represented by finite numbers. To control the processes of cellular, the expression level of genes evolves, is calculated by these interactions. A Biological Network that consists of a complex process has several nodes and these networks grow by regulatory entities. The relationship between proteins and genes is summarized by given network. Traditionally, the systems of biological modelling have been accomplished by a system of Ordinary Differential Equations (ODEs) or Stochastic Differential Equations (SDEs) by biological circuit's connections and taking into account of entities. The concentration of gene product is represented by each feature in an ODEs' system, and non-linearity varies to another. A mathematical challenge is posed by the non-linearity of these ODEs and the development approaches of qualitative modelling are paved by this limitation [6].

1.3 Existing tools and frameworks

Many tools are being used for exploring biological networks which include GINsim, SM-BioNet, Pathway Studio, Cytoscape, Patika, and VisANT. For development of the system biology and integrative biology, these tools are very important. The emerging

trend for the development of biological network tools is to use the more dynamic model instead of 'static' representations of cellular state [7].

For the analysis and construction of logical models of multi-valued, GINsim (Gene Interaction Network simulation) is used and it is a Java based application. For the simulation and modelling of Genetic Regulatory Networks (GRNs), GINsim is used. The logical formalism of multilevel is performed by this tool. State Transition and Logical Regularity Graphs are supported by this tool. In functional genomics, modern developments have produced a huge size of data on gene expression and the underlying regulatory mechanism. Usually, these networks include a complex path that may have more than one feedback. For this purpose, simulation and modelling tools are an important complement to the experimental tools. As detailed and accurate information on mechanisms of molecular and also kinetic parameters' values are presently tough to determine. The qualitative methods are a good approach to analyses and model the essential and necessary properties of GRNs.

For understanding the organisms functioning on a molecular level, it is important to know which genes are expressed, where and when in the organism, and which extend. GRNs systems structured are used to achieve the gene expressions' regulation by interactions' networks between RNA, small molecules, DNA, and proteins. Most of the GRNs of interest contain several elements connected over interlocking negative and positive feedback loops. So, formal tools and methods for the simulation and modelling of GRNs will be essential [8]. GINsim contains a qualitative models' simulator of GRNs supported distinct, logical formalism. Through GINsim, the user is allowed to specify a model of a GRNs in respect of multivalued logical functions, asynchronous, and to analyses and/or simulate its qualitative impulsive behaviour [9].

SMBioNet tool is used to model the systems of biological regulatory. This tool is established based on the Computational Tree Logic (CTL) [10] and the multivalued logical formalism of Rene Thomas.

The input file comprises four sections, VAR, REG, PARA and CTL. The input file is divided into four sections. The first section (VAR) represent the variables, the second section contains REG the regulatory process of what we call a discrete regulatory network. The logical parameters in the third section, PARA. on the logical parameters associated with the network are given. The last one Computational Tree Logic (CTL),

contains contains a formula that represents a dynamic property which is then tested on all transition states.

GenoTechE is another java based graphical tool which is used to analyze the dynamics of given BRN. It calculates the state graph by using René Thomas' methodology. It can also be used to identify the diseased states and equilibrium states in the state graph. The states may be visualized in this tool for better understanding. GenoTechE supports the "dot" format to import and export graphs [11].

1.4 Limitations

For the last few decades, researchers have been working with these tools. These tools help us a lot in problem-solving as well as the experimental purpose. But we have to face some kind of limitations in using these tools which are mentioned below:

- The tools that are already being used by an individual do not provide a programming interface. Existing tools are not extendable and can not be modified as an individual to solve any other problem other than the already given i.e the biologists need APIs to solve the problems like parameter estimation.
- Ease of use: Existing tools need to be set up before use. We also need a training session to train the end-user before using it. So, there is a need for such a tool that is available online and ready to use at any instance. And that tool must be easy to use and understandable by the end-user. For this purpose, there must be a tool that has a considerable user interface so that the user may interact with the tool more conveniently.
- Interpretability: The file exported by the Cytoscape can not be imported by GINsim. The user has to rewrite/convert these files. Given that, the programmer must have to develop a different code to convert the files exported from GINsim to import in GenoTechE.

1.5 Problem Statement

Modelling, simulation and analysis of biological systems is a well-known problem in systems biology. The pathways are abstracted in the form of biological regulatory networks,

often represented as weighted directed graphs. René Thomas' qualitative modelling framework is a widely used technique to analyse biological networks and their dynamic behaviours. However, not much has been done towards the development of application programming interfaces to facilitate the development of packages and libraries based on the aforementioned framework. In this study, we address this problem by developing a programming framework in python language. Our work provides a well-defined programming interface to model, simulate and visualize biological networks using a qualitative modelling framework. The availability of the programming framework in python language makes it easy to use and integrate with existing packages.

1.6 Objectives

This study focuses on

1. Python package to support further development of systems biology applications based on the qualitative modelling approach of René Thomas'
2. Development of a web-based tool for modelling, simulation and visualization of biological regulatory networks using qualitative modelling framework.
3. Cross-platform implementation to ensure portability across existing tools such as GinSim, SMBioNet, GenoTechE, etc.

Chapter 2

Literature Review

2.1 Modelling of Biological Regulatory Networks

Biological Regulatory Networks (BRNs) represent the interconnections between biological entities. The graph of the BRNs consist of the regulatory components. The interaction of cellular networks comprises of genes, tiny molecules, proteins, their mutual interactions, DNA, RNA, and their complexes. The link captured in the gene is first recorded into messenger RNA (mRNA) when a gene is turned on and then respective link is interpreted into proteins. Proteins subset has a monitoring role which is able to turn other genes on/off. Such regulations among genes and proteins may involve difficult operations in the form of a network that leads us to the Biological Regulatory Networks (BRNs).

BRNs are required to be investigated, for experimental manipulation and for an understanding of how BRNs work. Biological Regulatory Networks (BRNs)' formal methods are recommended since 1960 for investigation. Several protocols have been developed to create behavioural expectations as of the model of the framework utilizing examination (based on simulation) as well as to build the model by using investigative data-set on monitoring mechanisms. Few proposed protocols, along with their techniques for simulation are being discussed here. The new protocols consist of stochastic equations, directed graphs, differential equations (ODEs and PDEs), Boolean networks and their generations. The modelling framework is categorized into four sections by using the proposed protocols that include different types of modelling; quantitative, qualitative, hybrid and piece-wise linear differential equations (PLDEs).

A Biological Regulatory Network (BRN) is analyzed as a group of molecular objects. These entities interact collectively with one other. The biological objects are presented by nodes in the network. The interactions are represented using edges that show activation or inhibition of that entity. There are numerous monitoring networks involved to capture segments linked through meshing positive and negative loops for feedback, and it becomes difficult to analyse the interactions. Resultantly, the proper approaches and tools used for modelling the simulations, used computer techniques in genetic regulatory networks that can be crucial. Each node in BRN is demonstrated as a unique fickle which has a limited number of likely outcomes. BRNs consists of complex processes. A connection between genes or proteins is summarized by the network. Usually, demonstrating of biological schemes is done by the use of Ordinary Differential Equations (ODEs) or Stochastic Differential Equations (SDEs).

Over the two years, proven verification is raised as an important method for conventional demonstration and investigation of ongoing responsive and capricious frameworks. The key benefit of the model is examining an analysis based on simulation and its characteristics and consistency of calculated results. They clea state model regulator (SPIN) for proper modelling, and for examination, Linear Temporal Logic (LTL) of the complex dynamics (cycles) of BRNs was used. A framework was modelled for BRNs on the Kinetic Logic of René Thomas, and a state of the art, SPIN model regulator. The framework is based on boolean modelling and can also be used for the analysis of BRNs objects. To check the quality and practical usefulness of the structure, and for analysis BRNs of pseudomonas aeruginosa and Indoleamine 2,3-dioxygenase is cast-off [12].

Gene network, represented by completely associated Boolean systems, where every component communicates with all components including itself. In this method, an input circuit creates its normal dynamical behaviour (i.e. multistationarity or oscillations) just to fit under the estimation of logical constraints. After every constraint, the state is known as functional. This formalism permits the calculation for the constraints, applied on the logical limitations to know whether the input circuit is useful or not. From a biological perspective, it recommends that the formal regulatory networks can be degraded to smaller but independent feedback circuits. In other words, this connection expresses that the bigger the circuit, the greater the arrangement of limitations to be fulfilled [13].

The Petri Net (PN) formalism proposed an integral structure for examining the dynam-

ical behaviour of big systems formed by qualitative or quantitative understandings. It is a PN formalism with a logical approach, Which is difficult and systematic plotting of multi-level logical regulatory models into exact standard Petri nets, called Multi-level Regulatory Petri Nets (MRPNs). On the other hand, some reduction strategies are also proposed. This consolidated approach includes two main phases. First, the specification of the model is done regarding a basic regulatory graph, tracked by its parameterization. The flexibility provides an advantage over the logical limitations and then MRPN corresponds to the resultant systematically generated graphs and lets the application of the current techniques to assess the dynamical property [14].

An integrative approach for verification of regulatory networks based on the biological study analyzes that the previous verification methods suffer from different issues i.e., different characteristics of network dynamics like fluctuations that are not expressed easily utilizing classical logic, mostly the existing set of tools are unable to perform such evaluation. In order to reduce these limitations, it had been proposed that CTRL logic that is an extension of CTL, with fairness operators and regular expressions. The method permits a natural description of biological properties i.e., multistability. This method aims to offer a syntax that is simpler to benefit those users who are not experts to formulate different complex biological problems [15].

A network based on data that is regulatory gene-to-gene microarray obtained by inference method from yeast. This particular method finds the out-degrees by studying the simplified network. The nodes with extraordinary out-degrees are unlikely but effective from a biological view. The study also finds a biological group of genes that are relevant among many of the genes and also the excess of genes coding for different products included in the record between genes with higher out-degree. Simulation results proved that usage of linear approach is valid and efficient for getting large-scale networks [16].

2.2 Quantitative Modelling

Quantitative modelling methodology relies on two different equations; The ODEs and The PDEs that are being used tremendously to investigate BRNs. BRNs are generally represented by biological experts in the format of directed graphs or logical feedback circuits [17, 18]. The dynamics of these intuitive circuits and graphs are then investigated

for deducting the elements from the frameworks. The simplest and straight way is to depict a network in the form of a regulatory network. Previously, the formalism of directed graphs was utilized to provide an example of the regulatory networks. A directed graph G consists of nodes and edges which is represented as a tuple $G < N, E >$, where N is the set of nodes and E is the set of edges. Moreover, a directed graph is defined as a set of vertices $< i, j >$, where i signifies the head and j signifies the tail of the edge. The biological objects are exemplified as nodes of the directed graph. The edges indicate the interactions among the biological objects (nodes). The kind of formalism is summed up in a few different methods. The edges and nodes can be marked to denote the rate of the interaction of biological objects. The directed edges can be expressed as a tuple like $< i; j; s >$, where i denotes the head of the head j denotes the tail and s denotes whether i is activated or inhibited by j. Limitation of differential equation-based modelling is that it requires enormous CPU time and memory along with the inherited incomplete nature of numerical simulations [19].

The respective study showed that Petri net permits the combination of analyzing qualitatively and mathematically. The findings were responsible for the reduction of essential limitations to fulfil system performances noticed in monitored networks for a gene [20]. The benefits of using hybrid Petri net models are: 1) The interface is user friendly interface that permits an easy simulation, visualization and design. 2) In continuous and distinct events, the model can handle the metabolic process and gene regulation. 3) The inhibitor arc is beneficial for studies to learn about enzymes interaction with a substrate to know about the role related to inhibitors in gene expression. The benefit of mathematical equations is that the simulations of the model are dynamically visible. The complex networks can be managed with the similar set of behavioural and structural properties. When HPN is used for such large network, the hierarchical assumption makes it easy to make a generalized version of HPN. HPN is abbreviation of? Write complete word after word when and then HPN should be written in brackets. A grained model for transcription of gene along with developed approaches to recover them from expression data of the gene within the model of a probabilistic approach is a great phenomenon. This method focuses on quantitative transcription rates and at the same time, it finds the kinetic limitations that are used to regulate these rates and the level of activities of regulators that are not observed. The proposed approach is applied to expression data-sets taken from yeast sample and prove that the scheme is capable

to learn the dynamics of unknown activities. The scheme also proposed a new learning technique and shows its accuracy [21] .

Different approaches for both logic and quantitative modelling of gene and molecular networks were studied. During the process, Most examples were considered from gene networks. The study led to the way for researchers to make a contribution in these fields and have better knowledge of these. Mathematical modelling is an essential part of biological systems. Well-designed and evaluated models help to learn the cellular and molecular process and can find the causes of mutation and drugs. The study gave awareness of various methods and showed ways to combine these models for making them more useful.[22]

Quantitative modelling technique based on fuzzy logic to manage kinetic data that is unknown and further produces related results in case kinetic energy is not complete. This study successfully evaluated the proposed model with the combination of the ODE relied model. The proposed biological model was not only dependent on the information, not related to kinetic data, but also on known kinetic information. Variance among the conventional and fuzzy model will increase by the process numbers that is demonstrated with uncertain logic. However, the technique is capable to show quantitative output with biological relation [23].

A model which covers the space among the frequency running in background of the nucleotide and rate of gene gap is recorded. The record model presented that it depends on energy binding among regulators and their targeted genes. The regulatory efficiency, kinetic parameters and binding affinity are used to get information of features related to the promoter are modelled as a binding energy function. The un-observed transcription factor and kinetic functions are not incorporated into a probabilistic technique. Experimental outputs that depend on the data-set of yeast reflects that the method can effectively find the regulators attention [24].

2.3 Qualitative Modelling

It is the simplest and fundamental modelling approach. It is distinct and logic-based technique, firstly introduced by René Thomas and Kauffman. Qualitative approaches include Boolean login and Kinetic logic of Thomas. In 1970s, Kauffman et al. demon-

strated the activation levels of objects with Boolean variables in BRNs. The Boolean variables comprised of only two promising states which are "1" and "0". These states can also be represented as "ON" and "OFF". The logic based Boolean methodology is suffering from several challenges and limitations in analysis. It is only dealing with two levels, however observing the difficulty of biological objects, their modelling requires more than the two-levels for apprehending their innate dynamics precisely. Kinetic theory and René Thomas' model are multi-valued logic that permits us to estimate the sigmoid nature of biological entities more closely. This approach is observed as procrastinating in a distinct domain. Whereas, the expression level of a gene u at time $t+1$ is ruled by the function, involving equations from the regulators of u. René Thomas et al. prolonged Kauffman's contributions to the multi-state models. In multi-state methods, the expression's level of some biological object in BRNs is modelled by a distinct variable [25].

The basic dynamics of Rene Thomas models are overseen by standards of model parameters within a due time, also known as logical parameters. Mostly, the actual measurement influenced by biological entities cannot be exerted. Therefore, discovering suitable model parameters that are going to explain tough regulatory controls in qualitative modelling that are used in inter-cellular processes in biological systems. Although, when modelling complex disease progression in different scenarios, it is essential to calculate the parameters that can guide the system to a recommended performance level/scenario. The computational model can predict the therapeutic intervention. A new technique to find the unidentified logical parameters by using a formal authentication method, known as model checking was also proposed. In these frameworks, the semantics of BRNs are specifically modelled like some state-transition system. Every state model has a conceivable formation of requisite system that may possess numerous promising successor states. Model organizer calculates the state space of the bio-network, by gaining a Cartesian product, among the thinkable configurations of the system. Hypotheses are experimented in the temporal logical equation that is developed by logical links and operators [26].

A technique constructed on the model and analyzing the Regulatory-Network of Indoleamine 2, 3-dioxygenase in Tumour Immune Escape, where Immune discharge is a serious entrance to malignancy was practiced. According to the findings, to suppress the T-cell immunity in pathological settings Indoleamine 2, 3 Dioxygenase (lDO) is engaged

including cancer. To demonstrate the BRNs utilizing kinetic logics of Rene Thomas, a renowned and dominant method for modelling formalism, provision to comprehend their dynamical performances of the IDO related BRNs. The homeostasis of IDO and T cell mediated protected regulation in the body that is accountable for overpowering the tumour by monitoring IDO activation, is revealed by a qualitative cycle. A distinct modelling of the IDO associated BRN was used. IDO attentiveness in a cell ought to be under constricted control, by keeping up the homeostasis of the framework communicating IDO. This BRN can be additionally tuned when postponing parameters that are being investigated by utilizing the hybrid model examination devices [27].

The consolidated utilization of numeral-valued logical formalism, and, distinct Petri-nets to demonstrate and analyze regulatory networks qualitatively was summarized. This explanation reflects on a distinctive qualitative framework legalized by non-existence of specific quantitative data regarding regulatory mechanisms. Logical process firstly presented by René Thomas and coworkers have verified that it is beneficial and well-equipped for the qualitative modelling of respective regulatory networks. It has performed very well for a variety of cellular mechanisms. Besides this, Petri nets (PNs) establish mathematical framework to characterize distinct simultaneous schemes. Since a huge quantity of formal effort and computational advances are achieved, PNs' value from a well-developed mathematical framework and a group of suitable delays, dedicated computational techniques/tools proposed a methodology that is dependent on several workflows had been applied for a decent variety of cell forms. Then, the second one is Petri nets (PNs) which comprises a mathematical framework resulting in a methodology that is dependent on several workflows systems [28].

A well-developed Systematic Perturbation Qualitative Reasoning (SPQR), for reasoning qualitatively to systematize the acceptance of findings by systematic perturbation tests was evaluated. This technique is built on a qualitative concept of the investigative dataset. It is worthwhile to mention that for an agitation experiment, estimated standards of the experimental variables are lower, higher, or equal than the estimation in an irregular form, when no disturbance is conducted. This scheme uses a set of $IF - THEN$ rules for concluding fundamental affairs among the variables, examining the patterns that are broad casted for the disturbing signals via bio-network. This algorithm is especially designed to overcome the false rate of positive between the contingent relations [29].

A methodology, dependent on several work-flows, where diverse programming techniques combined with delicate parameters are bound together, with additional manual steps most of the time is a steady approach. To access and recreate such an approach is thought, as distributions are habitually neglected examination realities, and because of a portion of the instruments are tricky to introduce, and additionally have a sharp expectation to absorb information. The CoLoMoTo Interactive Notebook gives an insight for a unified domain to change, implement, share, and replicate analysis of biological networks' qualitative model. This system has advantage over various technological skills to guarantee the iteration and to decrease the user learning curve of these skills. The computational flow is applied by using the Jupyter web interface that enables textual annotations and uses clear code for execution and illustration. The consequent files were again executed in a particular environment. Till now, the CoLoMoTo's Interactive Notebook delivers admittance to the various software tools like GINsim, BioLQM, Pint, MaBoSS, and Cell Collective, for analysis and modelling of Boolean as well as numeral value bearing networks. In future, more up-to-date tools may be added. A Python interface was established for each tool that is proposing unified incorporation in the Jupyter web-interface, favours ease in connecting complementary analysis [30].

2.3.1 Parameters Estimation

Models are deliberations of real systems, and the qualities which are responsible for deciding how close a model is to a real system are known as parameters. Estimation of parameters is a difficult assignment since it is hard to get the right ones from the tremendous arrangement of parameter values. In like manner, parameters estimation of biological regulatory networks to study the dynamical changes in biological systems is as challenging as it is significant. Parameters estimation by utilizing qualitative modelling framework is favoured over quantitative modelling as articulation levels are represented using distinct qualities, which diminishes the trouble for parameters calculation job keeping the parameter state space limited; whereas, in the case of quantitative modelling where expression levels have continuous values, the difficulty for parameters computation is high as parameter state space is infinite. Also, the qualitative modelling framework unveils important properties of Biological systems i.e., bifurcation points, cyclic behavior in the form of feedback loops and stable or steady states, etc. Parameters estimation

isn't just useful in the investigation of systems-level diseases like Parkinson's disease, Alzheimer's disease, and cancer, etc., yet additionally advantageous in the identification of potential helpful medication targets. According to René Thomas Logical Formalism, the dynamics of a biological system are studied by converting its regulatory network into a state graph based on logical parameters. These logical parameters are unknown and a huge challenge for Qualitative Modelling is their estimation.

Model Checking is an automatic method for the confirmation of compound peripheral systems [31]. At first, they established for simultaneous program verification, the model examination is working as an industry-standard methodology for demonstrating the correctness of digital circuits, security protocols, and embedded systems. Model-checking methods are distinguished based on the way, they understand the notion of time either it is Linear [32] or branching [33]. Computation Tree Logic (CTL) is favors the expression of properties of non-deterministic dynamical systems such as BRNs, where an existing state can have more than one descendant state.

BRNs are accountable for developing and maintaining the inter-linked programs and functions in the organisms. The functions are applied by the dynamic nature of BRNs and are delicate to the regulations that stand compulsory by specific activators as well as inhibitors. The reasonable modelling formalism of René Thomas integrates the compassion of collection of logical constraints modulated specifically by existing regulators, changing over time. Due to an increment in the difficulty of BRNs, in case of the number of entities, their communications, the undertaking of parameter estimation turns out to be computationally costly with the existing sequential SMBioNET tool. The authors spread the current consecutive usage of SMBioNET by utilizing the data decomposition method with the use of a Java messaging library known as MPJ Express. The methodology separates the parameters space into various areas and then explored in a parallel way on the High-Performance Computing hardware (HPC). A qualitative modelling framework is extensively used for exploring the nature and functions of biological regulatory networks. Though, the calculation of model parameters in qualitative modelling is an intensive task with respect to computation. It almost offers linear speed-up on both clusters and multi-core platforms. Additionally, the parameters similarly recommend a potential therapeutic intervention that restores homeostasis [34].

2.3.2 Challenges in Qualitative Modelling

For validation, the modelling and simulations are made that require tools for model validation. The direct threats opposed to model validation are accomplishments that are based on the model experimentation and predictions of the data and also have a look at observations and comparisons. One of the big issues faced, was the development of a biological model that is lacking the graphical tool impact that manages the large networks. Qualitative models of dynamics of signalling trails and gene regulatory networks, allow to have chronological properties of bio-networks while demanding few parameters. Though, these distinct models typically agonize from the state space explosion problem that creates formal evaluation of their potential performances are pretty competitive [35]. Generally, qualitative models suited the system under constraint information regarding qualitative facts and then quantitative models are used to give details in a complete manner. Qualitative models do not incorporate usually, kinetic aspects of cellular signalling and can thus not provide a complete quantitative understanding as with mechanistic ODE modelling [36]

2.4 Hybrid Modelling

Hybrid modelling is comprised of a mixture and a combination of approaches. In the field of biology, it is said to be the coupling or combination of qualitative and quantitative formulation. Traditionally, the reaction-diffusion system of partially different Differential Equations, demonstrating the developments of chemical concentrations/densities, are united with the cellular auto-mata or available agent-based models. The properties of the biological characteristics change altered with continuous external chemical change. Most models follow the same pattern. Summarily, a hybrid model agrees to any interaction or coupling between the two models, which are not executing on similar pattern. For instance, deterministic and stochastic, global and local, phenomenological and physically based, etc [37]. A novel hybrid method for automated formal analysis of biological systems is established with an emphasis on their oscillatory behaviours, which allows incomplete and practical knowledge based biological data-set. To meet the needs of quantitative information, our modelling focuses on (i) the biological compound product signs and (ii) the temporal properties related to the biological properties of the

mutual interactions. Such limitations are informal to be removed from the available experimental data. They propose perceptive on a hybrid system, which is adapted to large gene regulatory networks, that is appropriate for underlining the biological properties [38]. Hybrid Systems and hypothetical models. Evaluations are recognized related to the data requirements, scalability along with network size, and computational burden. The visualized programs along with extensive case studies at huge scale genomic models & in particular sub-systems of various organisms [39].

The phenomenon of hybrid refers to the composition of two innately different things. In biology, the hybrid modelling is known as the combination of different continuous and discrete formulations having discrete characteristics. Traditionally, reaction-diffusion systems of Partial Differential Equations (PDE), reflecting the evolutions of chemical concentrations or densities, are coupled with the cellular auto-mata/agent based models. The state and/or characteristics of these entities, are evolved with the external continuous chemical any other fields. It is a fact that many models related to bio-medicine follow this pattern, hybrid modelling can be much more than that. Summarily, a hybrid model reflects any interaction or coupling between two or more models, and their formalism is entirely different. For example, deterministic and stochastic, global and local, phenomenological and physically based, etc [37].

A new hybrid technique for automatic formal analysis of biological systems is developed with a special emphasis on their oscillatory behaviours. It allows the use of incomplete and empirical biological data. To meet the needs of quantitative data-set, this methodology focuses on (i) the biological compound product signs and (ii) the temporal properties which are associated with the biological effects of different interactions. This information can be obtained from experimental data-set. They aim at reasoning on such a hybrid modelling system which is adapted to the large gene regulatory networks, which may be suitable for emphasizing biological properties [38].

Hybrid Systems and stochastic models. Comparisons are also established regarding different requirements of data-sets, their scalability, along with network size and computational needs. These methods of developing requisite models are incorporated with successful case studies in large-scale genomic models, and in the particular subsystems of different organisms [39].

There are a few tools available to model biological regulatory networks. Two of widely

used are SMBionet and GinSim their workings are as follows:

2.5 SMBioNet

SMBioNet [40] uses qualitative biological networks and quantifies them into a robust quantitative model. The tool uses Thomas' algorithm for the stategraph quantification. The first step in the tool is gather biological information including static knowledge [41], for instance about the gene, its sequence, and its ontology to populate node characteristic; the second ifnormation is the dynamic properties that changes as a result of the weights of the node structures, the influx and regulation of the gene or protein as a result of its interactions with its environment. These dynamic properties include whether the stategraph is homostatic or multistationary etc [26]. The next step is to formalize these inputs into a a node structure and its associated edges with direction. Thirdly, a way to hypothesize about the given nodes and its interactions are proposed. These hypotheses can be reiterated until appropriate hypothesis is developed. In comparison to this.

2.6 GINsim

GINsim [42] is another tool that translates biological networks into stategraphs. Thi tool provides a graphical user interfae using JGraph, an open source Swing tool. Regulatory networks can be input into the software for analysis. After the input, simulation can be run. A separate window uses different weights that can be input for the simulation where nodes activate or repress the other through an interaction. For a simulation of n nodes can lead to a stategraph of 2^n states. For optimization, the tool uses depth first search to simulate on a given node. Subsequently the more important states on genes are prioritized by the user.

2.7 Research gap analysis

We need a tool with user interface that is user friendly and open source, so that anybody can modify it according to their requirements. For existing tools, we need a training session initially and they are not user friendly. So, we need a system that can be modified by the user, according to the individual's requirements and must be user friendly. The

other issue in the existing scheme is that the file exported by the one tool cannot be imported by the other. The user must rewrite/convert these files. We need a package for system biology that is based on qualitative modelling approach. There is a need for a tool which is web based and provide the modelling, simulation and visualization of biological regulatory networks.

Chapter 3

Methodology

3.1 Overview

This chapter discusses the methodology to develop the approach used in this work. Given the popularity among data science and bioinformatics researchers, Python language is used to implement René Thomas' approach. From that BRN and its parameters, next dynamic states are computed and are used for the state graph generation and then analysis can be performed on generated state graph. Figure 3.1 shows the workflow of the approach. Every step is discussed one by one in detail.

3.2 Systems Biology

Systems biology is the mathematical analysis and computational modelling of biological systems. It is a biology based interdisciplinary field of study that focuses on complex interactions within biological systems. The main idea behind systems biology is that to understand biology, it needs to be studied on the systems level, considering gene regulatory networks, protein interactions, or metabolism networks as a whole. Kauffmanś qualitative modelling approach represents the activation levels of different entities in a BRN by utilizing different Boolean variables which are possessing only two possible states i.e. "ON (active or 1)" and "OFF (inactive or 0)" also called Boolean Modelling. Initially, René Thomas' proposed a qualitative framework based on Boolean logic which can be widely applied on BRNs which precisely approximated the different ODE models. Later, it was widely admitted that the Boolean model is inefficient to completely reflect

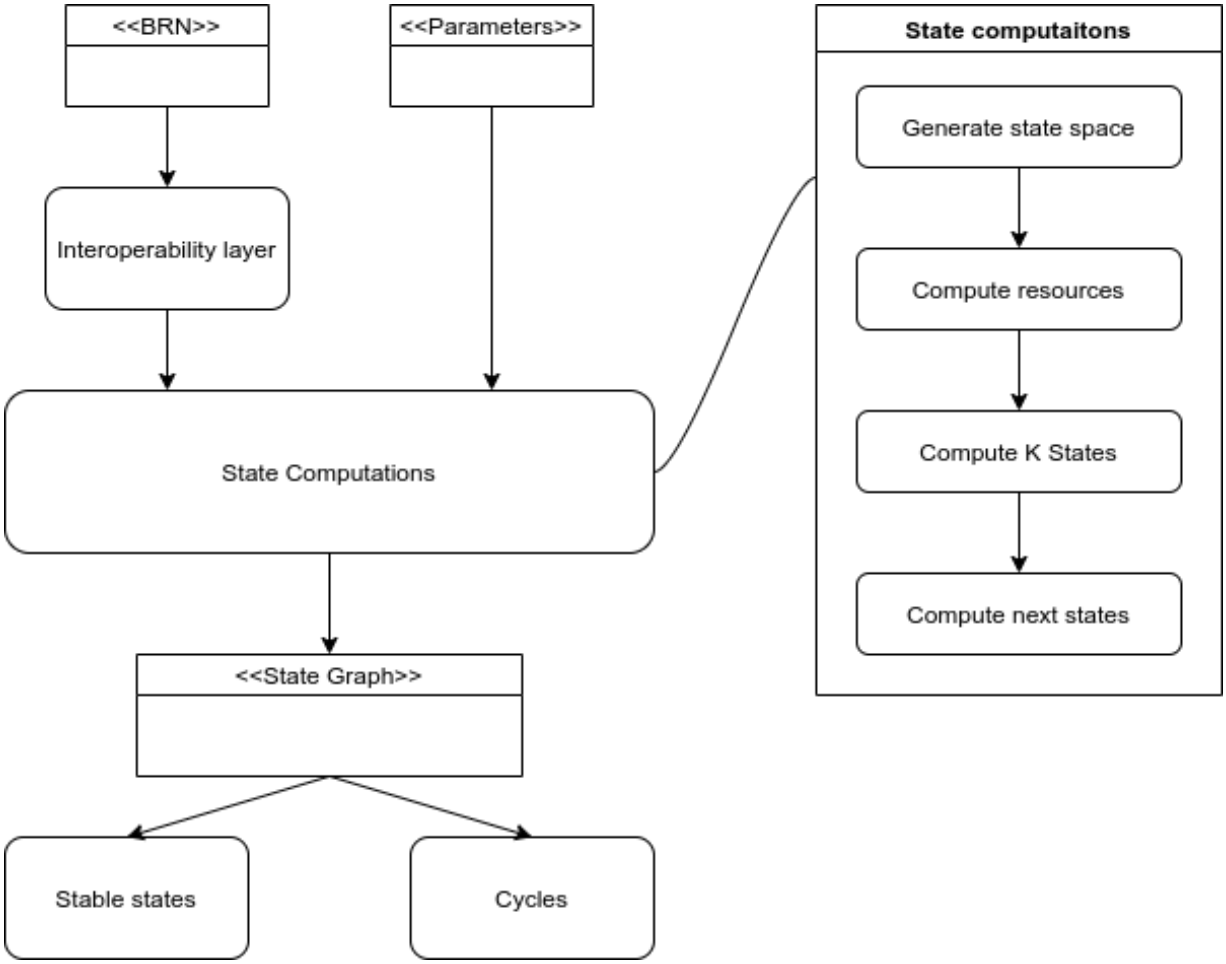

Figure 3.1: Workflow starts by a BRN as input with parameters, which is processed and in the end, a state graph is produced. Interoperability layer provide an easy to use API for conversion of graph formats.

different interactions within BRNs, at different gene expression concentrations. This led to the presentation of kinetic logic formalism which allows the modelling of discretely abstracted concentration levels other than Boolean as well. Qualitative modelling approaches have been used to model the behaviour of several biological networks including MAL associated BRN, dengue virus pathogenesis and clearance mechanism and many more. Thomas' Formalism is primarily based on modified graphs, known as Biological Regulatory Networks (BRNs).

3.2.1 Pseudomonas

Pseudomonas aeruginosa is a species of bacteria that live in the human lung. The genes under study here encode mucus production which severs symptoms in cystic fibrosis

patients. This AlgU gene positively regulates an operon and all genes involved in mucus production including AlgU itself. This is a classic An operon is a set of inter-linked genes that are translated to protein together. case of positive auto-regulation. The model presented here is simple but it can accurately encapsulate and give insights into potential therapeutic solutions [25]. The problem under consideration here is how can the mucus production network be translated into a state graph that can be modelled through BRN. In Figure 3.3 we have a biological regulatory network of Pseudomonas aeruginosa where x represents AlgU and y represent anti-AlgU gene respectively.

3.2.2 Malaria

The second pathway under study is that of malaria. Malaria is caused by the Plasmodium parasite that is trasnmitted from one organism to another by the Anopheles fly. Symptoms of malaria include chills, elevated body temperature, and in severe cases death. The Plasmodium parasite feeds on the host body's red blood cells and ruptures the cells to come out. This causes a severe drop in the number of red blood cells in the body causing death if not taken care of. Symptoms are often severe in children. Chloroquine phosphate is used to treat patients with the disease.

In this study, a special case of malaria is considered called cerebral malaria. This is the case when red blood cells going to the brain are heavily sequestered which limits the amount of oxygen supply with a simultaneous increase in nitric oxide. This leads to coma in a condition called encephalopathy or cerebral marlaria. To understand this pathway, in this study, the BRN of MyD88-adapter-like (MAL) and its dependent proteins were used to model hyperinflammation, or the abnormal increase in body temperature. Another study showed that MAL, through different snignle nucleotide polymorphism studies, is involved by being activated by the Bruton's tyrosin ekinase (BTK) through phosphorylation [43, 44]. This started an inflammatory response by activating cytokines as well as SOCS-1 (suppressor of cytokine signalling 1) protein which degrades phosphorylated MAL and acting as a negative feedback. These three proteins are modeled here through BRN using the canconical algorithm.

3.2.3 Dengue Virus

The dengue virus is a single stranded RNA virus that has four serotypes that can equally infect the human body through the Aedes aegypti mosquito. Serotypes refer to the sub-classes within the dengue virus family. Dengue is caused by the bite of the Aedes aegypti mosquito which transfers the virus through saliva to skin cells which then trigger an inflammatory response throughout the body. During the viral incubation phase, the virus can induce fever, fatigue and severe muscular and joint pain. In extreme cases, the virus can elicit haemorrhage inside the blood vessels that causes internal bleeding under the skin and abdominal pain among other symptoms. This virus is found in the tropical and sub-tropical regions of the world. In the present case study, viral binding to the Toll-like receptor 3 that induce innate immunity is modelled. Dengue virus uses the SOCS protein to inhibit innate immunity by binding to the Toll-like receptor 3 proteins. This is done by using the Rene Thomas formalism to capture the effects of the interaction between Toll-like receptor 3 of human cells and SOCS protein in dengue [45].

3.2.4 Hexosamine Biosynthetic Pathway (HBP)

The Hexosamine Biosynthetic Pathway (HBP) is a component of the glycolysis pathway that is involved in protein glycosylation. Protein glysylation inolves the addition of glucose groups to the protein moiety which stabilizes protein for further activity [46]. Disruption in HBP has been implicated in cancer where tumorous cells go to metastasis, which is when tumour cells migrate from the local center to the blood system and circulate throughout the body. Therefore modelling HBP and its rather complex network is very relevant to cancer genomics. HBP is yet to be well-understood because of how it is intertwined with the glycolytic pathway. But the rate limiting enzyme, which is the slowest and most important step in a chemical reaction, is the protein Glutamine-Fructose-6-phosphate amidotransferase (GFAT). This pathway involves the addition of N and O-glycosylation [47] which means adding a glucose group to target protein's Nitrogen and Oxygen atoms. What is not understood in the pathway is how these phenomena affect the body's functions, but several studies have shown that GFAT1 and GFAT2 are involved in eliciting many different types [48–52] of cancerous events, including metastasis. These genes are also involved in exacerbating other tumours [53–

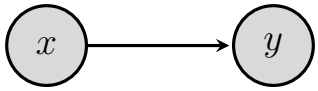

Figure 3.2: Directed Graph representing nodes x and y with and arrow showing the direction of the edge.

55]. Additionally, studies have shown that increase in HBP leads to hyper N and O-glycosylation events that reduce cell motility and therefore melanoma cell migration. Here is this study, HBP is used to understand this, owing to its breadth of complexity in terms of networks.

Qualitative modelling by using René Thomas' approach is a widely used approach for Study of BRNs However, the availability of the framework as an API and open-source implementation remains a challenge to date. Therefore, the present study is aimed to develop a qualitative modelling framework based on Python; a programming language widely used in the development of computational life sciences applications. Later on, the developed programming interface may be integrated with existing system-biology applications to have more efficient usage of existing systems

Following are the definitions explained in [56] which are fundamental to this work. The intent is to give the reader an overview of basic concepts.

Definition 1 (Directed Graph)**.** *"A directed graph is a form of a graph that is is defined as an ordered triple $G = (V, E, f)$, where f is a function that maps each element in E to an ordered pair of vertices in V. The ordered pairs of vertices are called directed edges, arcs or arrows. An edge $E = (i, j)$ has direction from i to j".*

Directed graphs are suitable for the representation of networks describing biological pathways which show the sequential interaction of elements at one or multiple times. It also points and the flow of information throughout the network. These are mainly known as regulatory networks. In the theory of directed graphs, a path is a chain of distinct nodes, connected by directed edges, without branches or cycles. The graphs we are using in this study are directed weighted graphs. In figure 3.2 we have a directed graph with nodes x and y and a directed edge from x to y.

Definition 2 (Weighted Directed Graph)**.** *"A weighted directed graph is defined as a graph $G = (V, E)$ where V is a set of vertices and E is a set of edges between the vertices $E = (x, y)|x, y \in V$ associated with it a weight function $w : E \to R$, where R denotes*

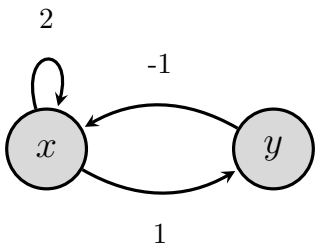

Figure 3.3: Weighted Directed Graph with weight/threshold specified on the edge. Edge from x to y has weight 1. Edge from y to x has weight -1 and edge from x to x has threshold of 2.

the set of all real numbers."

Most of the time, the weight w_xy of the edge between nodes V_x and V_y represents the relevance or strength of the connection. Usually, a larger weight corresponds to the higher reliability of a connection. Weighted graphs are currently the most widely used category of graphs throughout the field of bioinformatics. The weights on the edges represent the concentration required for a biological entity to activate or inhibit the other biological entity.

Definition 3 (Biological Regulatory Network)**.** *"According to the qualitative model of René Thomas, the biological entity represented as a weighted directed graph $G = (V, E)$ a node $i \in V$ may have multiple activators and inhibitors where*

- *biological entities are represented by a set of nodes V*
- *interactions are represented by set of a edges $E \in V \times V$*
- *Each edge V_i, V_j is labelled by a pair (τ, σ) where τ is the threshold at which gene V_i regulating gene V_j and $\sigma\{+, -\}$ is called a sign of interaction. ($+$ for activation and $-$ for inhibition)".*

Definition 4 (State)**.** *"The State of BRN is n-tuple $S = \{s_{v_i}, ..., s_{v_n}\} \forall s_{v_i} \in \sigma v_i$ where*

- s_{v_i} is the abstract expression level of v_i.
- In a given state, each s_{v_i} is regulated by its predecessors G_{v_i} formally denoted as set of resources (ωv_i)".

Definition 5. *(Resource) "The resources ωv_j of an entity at a state s_{v_j} resources gives the level towards which the entity v tends to evolve. Let $G = (V, E)$ be a BRN. The set of resources ωv_j at a state s_{v_j} is defined as*

$$\omega v_j = vi \in G^-_{v_j} | (s_{v_i} \geq \tau_{v_i} v_j \text{ and } \sigma_{v_i} v_j = +)".$$

Definition 6 (State Graph)**.** *"A state graph is a graph that represents how a biological system represented as a graph behaves given certain values of parameters. Let $G = (V, E)$ be a BRN and SV_i is the expression level of V_i in a state $s \in S$. Then the state graph $R = (S, T)$ is a directed graph, where S represents a set of states, and $T \subseteq S \times S$ is a relation between states also called the transition relation, such that $s \to s' \in T$ iff:*

- *$\exists$ a unique $v_i \in V$ such that $s_{v_k} \neq s^{`}_{v_j}$ and $svx^{`} = sv_i \Rsh K_i(\omega_{v_i})$*
- *$\forall v_j \in V\{I\} s^{'}_{v_j} = s_{v_j}$"*

Let $G = (V, E)$ be a BRN. The set of resources $\omega(x)$ at level s_{v_x}, is defined as:

X	Y	ω_x	ω_y
0	0	$K_x[y]$	$K_y[]$
0	1	$K_x[]$	$K_y[]$
1	0	K_x[y]	K_y[x]
1	1	$K_x[]$	K_y[x]
2	0	K_x[x, y]	K_y[x]
2	1	K_x[x]	K_y[x]

The parameter set is a Cartesian product of each entity's resources

K_x	0
K_xx	2
K_xy	2
K_xx,y	2
K_y	0
$k_y x$	1

In biological terms, the model above shows a simple system where node x and y represent two genes, are regulating each other in the following manner: gene x activates y with

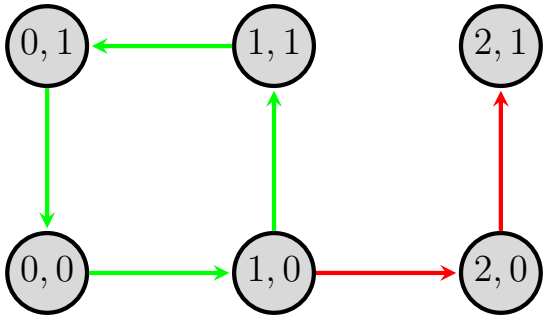

Figure 3.4: State Graph of pseudomonas aeruginosa for a given set of parameters. States $(0, 1)$ $(1, 1)$, $(0, 0)$, and $(1, 0)$ represent equilibrium while $(2, 1)$ and $(2, 0)$ states represent a deadlock.

a weight of 1, whereas y inhibits x with a weight of -1. These two weights are equal in magnitude opposite in direction, implying that the two genes equally excite and inhibit each other. There can also be cases when the inhibition is twice as high as activation. Additionally, gene x activates itself with a magnitude of 2. In other words, this model shows two genes that activate and inhibit each other with the same magnitude while the first gene activates itself with twice as much potential as it activates the second gene.

3.3 Implementation

Implementation is divided into three parts. At the core of this project is the implementation of René Thomas' algorithm, and perform analysis on generate state graph e.g finding stable and unstable states, then we have a module that allows the implementation to work with multiple other graph formats which are used by other tools e.g. GINSim, Cytoscape.

3.3.1 Graph Representation

As discussed in definition 2 we need to represent biological entities as nodes in a graph with additional information, e.g minimum and maximum thresholds of the entities can reach for this purpose we have used NetworkX a Python language package to represent graphs and analyse them [57].

3.3.2 Parameters

Parameters of a BRN are the thresholds/levels at which we want to see how the biological entity behaves. These parameters are the inputs given to each entity in a given time step. These inputs change as a result of the network's structure and transition function over time.

3.3.3 Interoperability Layer

As researchers can use multiple existing tools to analyse their BRN for example, Cytoscape works based on sif and graphml files while GenoTechE works based on Graphviz dot file format. To help easily convert between these formats we have implemented a compatibility layer that can import and export BRN to and from GraphViz dot, GraphML, Simple interaction file (sif) formats.

3.3.4 State Transition Graph

Generation of state transition graph is the core part of this work which is based on René Thomas' explanation of the state graph is given in definition 6.

To generate a state graph we first need to compute the state space for entities of BRN. The state space encapsulates all the combinations of states our entities under study can attain. These states may also transition between different states. A gene or protein may get activated or silenced by the activity of five other genes or proteins simultaneously. These states need to be modelled using the state transition graph in BRN. Transitions between states are represented by weights with a direction, whether the given state is activated, given a positive integer weight, or silenced, which is given a negative integer value.

3.4 Tools and Frameworks Used

For development of proposed PyPi library and web based user interface, we have used Open Source or freely available tools. To Write the whole code PyCharm Community version was used as integrated development environment.

3.4.1 Integrated development Environment

Integrated development environment (IDE) is a tools which facilitates the programmer to write source code. For development of PyRThomas and Web based interface of it PyCharm was used. PyCharm is is a widely used, feature rich development environment for professional Python development, although it is a paid tool but for educational purposes they provide a educational license to help students take advantage of all paid features. With student license we also got access to IDE's support for Web development which is essential for development of software proposed in this study.

3.4.2 NetworkX

To implement René Thomas' formalism we need to represent biological entities as weighted directed graphs, we found NetowrkX as best to be used for development of the framework because of it's high performance and maintainability. It also provide utility functions to perform some basic functions and analysis as well [58].

3.4.3 Angular JS

For the development of web interface we used Angular Js as a framework which allows us to dynamically manipulate webpages and change the behavior without refreshing the page and allow us to implement routing to allow user to navigate between the pages [59].

3.4.4 Cytoscape js

Cytoscape js is a graph network standalone library which is used to represent graphs in a web application. It provides support to show graphs in a very user friendly way. With it's canvas it allows user to interact with the graphs as well. It also support many layouts which can help biologist understand the behavior and dynamics of the network [60]

Chapter 4

Results

In this chapter, we highlight the results of our implementation of Python framework. With the help of datasets available in literature, we show how or framework can be used to model biological networks and computation of stable states and cycles.

4.1 Dataset

To test the performance of our developed framework we started with pseudomonas aeruginosa. Pseudomonads is one of the bacteria species that reside in human lungs responsible for cystic fibrosis. The gene *AlgU* controls mucus production by regulating the operons and all the involved genes. *AlgU* autometically regulates its expression in the system by making a feedback loop. We modelled this auto-regulated yet simple BRN and translate it into a state graph. The figure 4.1 shows 2 nodes and 3 edges where node X represents $AlgU$ and Y represents the $anti - AlgU$ gene [61].

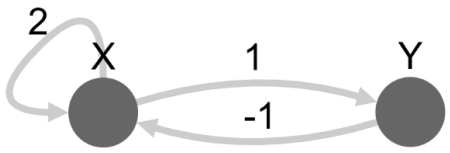

Figure 4.1: Biological regulatory network of pseudomonas aeruginosa.

Another pathway selected for study is malaria, caused by the parasite plasmodium and transmitted through vector fly Anopheles. The parasite attacks the host RBCs and comprises the immune system. Typical symptoms of malarial infection include high body temperature, chills fatigue and death if the number of RBCs drops in the body.

In a special case called cerebral malaria the red blood cells of the brain are sequestered which leads to the shortage of oxygen in the brain often lead to the patient death due to hypoxia. To understand the underlying molecular physiology of hyperinflation in malaria, the network of MAL and associated genes were modelled [41] as shown in figure 4.2.

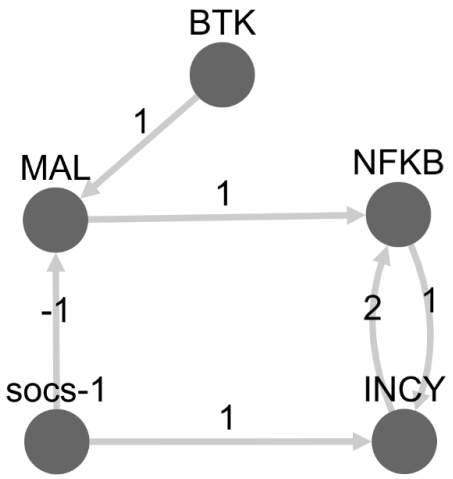

Figure 4.2: The celeberaial maleria associated biological regulatory network.

In addition, we utilized the dengue virus-related regulatory network shown in figure 4.3 for modelling using developed framework. Dengue virus is a single-stranded RNA based virus that infects the human body through the mosquito Aedes aegypti. Once the virus enters the human body it lives here using the host machinery for its replication and protein production. During the virus incubation time, the host body exhibit symptoms such as fever, body aches, joint pain and fatigue. In some cases, the dengue virus can cause haemorrhages inside blood vessels leading to internal bleeding that might worsen the symptoms [45].

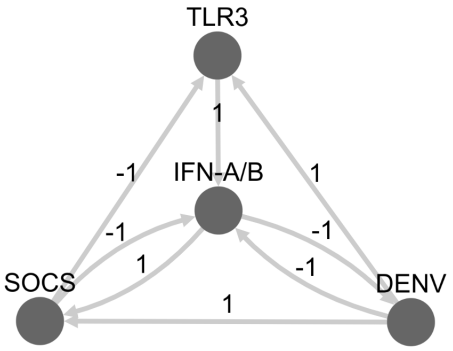

Figure 4.3: The TLR3 associated biological regulatory network of dengue virus.

The Hexosamine Biosynthetic Pathway (HBP) is one of the components of glycolysis which is responsible for protein glycosylation represented in figure 4.4. Protein glycosylation is an important post transcription modification that stabilizes the functional structure of the protein [47]. HBP has been repeatedly reported to be involved in tumour progression and cancer metastasis. The link between HBP and the glycolytic pathway is not well understood which makes the modelling of HBP relatively a complex task.

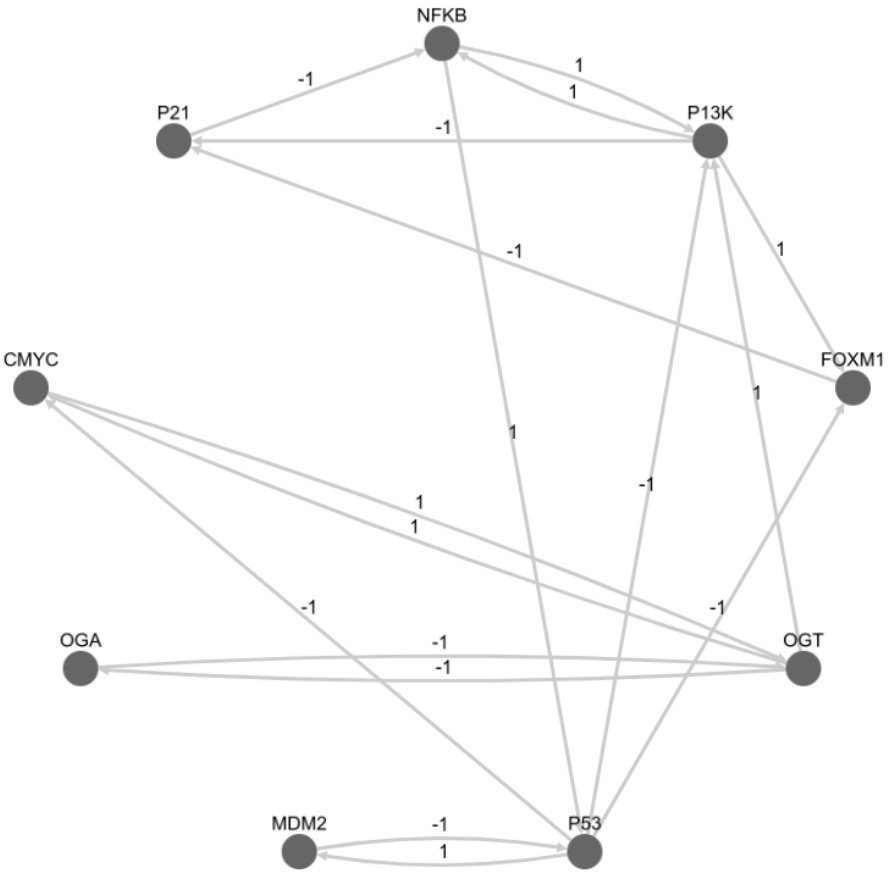

Figure 4.4: The Hexosamine Biosynthetic Pathway represented as a BRN.

4.2 Application Programming Interface

PyrThomas is split up in two modules one is NetworkService and the other one is NetworkAnalyser. NetworkService is responsible for representation of the network. It supports creation and deletion of nodes and edges from a network, clear the network, import and export network in sif, graphml and dot format. To support NetworkService there are two models defined, named Node and Edge. The Node represents the biological

entity with an additional properties to support René Thomas' approach. Additional properties are min and max which represent the minimum and maximum threshold that can be reached by an entity. Edge represent the interactions between the nodes, with additional property called weight which represents the threshold at which the source entity will activate or inhibit the target entity. The signature of the models (Node and Edge) is represented in 4.5.

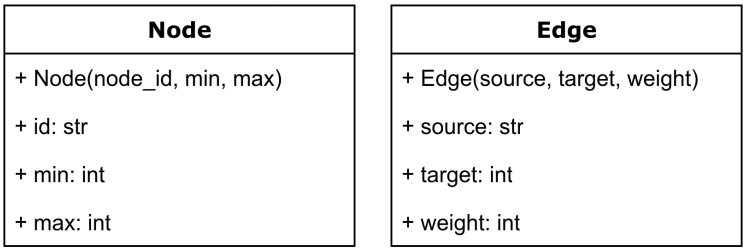

Figure 4.5: Node and Edge class diagram

All available method for NetworkService are represented in form of a class diagram in figure 4.6

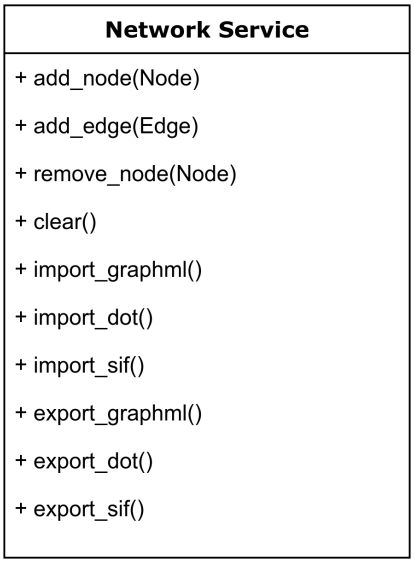

Figure 4.6: NetworkService class diagram. Using network service one can eithr create the network manually or import from a file. It can also export a given network to graphml, dot and sif formates.

The methods described here represent functions that can be performed within the pyRThomas framework. A brief explanation of each of the methods is as follows:

```
network: BRN represented as a networkx directed graph.
add_node: Adds a node to the graph
add_edge: Adds an edge to the graph with its respective weights
remove_node: Removes a node from the graph
clear: Destroys all the nodes and edges in the network
import_graphml: Imports from a graphml file as input
import_dot: Imports from a .dot file
import_sif: Imports from a .sif file
export_graphml: Exports data to a graphml file
export_dot: Exports data to a .dot file
export_sif: exports data to a sif file
```

Listing 4.1: Network Service API methods

4.3 Application on a case study

The case study chosen to be analysed under this developed framework is that of cerebral malaria 4.2. There are some proteins associated with MAL like BTK, INCY, NF$k\beta$ and SOCS-1. To model cerebral malaria using the framework developed we first need to install the package made available through the official python package manager called PyPI

Execute the following command to install:

```
$ pip install pyrthomas
```

Once pyrthomas is installed we need to import NetworkService, Node and Edge and then define the nodes and edges as given in listing 4.2.

```
from pyrthomas.network import NetworkService
from pyrthomas.models import Node, Edge
service = NetworkService()

# add nodes
service.add_node(Node('BTK', 0, 1))
service.add_node(Node('MAL', 0, 1))
```

```
service.add_node(Node('NKFB', 0, 2))
service.add_node(Node('SOCS-1', 0, 1))
service.add_node(Node('INCY', 0, 1))

# add edges
service.add_edge(Edge('BTK', 'MAL', 1))
service.add_edge(Edge('MAL', 'NFKB', 1))
service.add_edge(Edge('NFKB', 'INCY', 1))
service.add_edge(Edge('INCY', 'NFKB', 2))
service.add_edge(Edge('INCY', 'SOCS-1', 1))
service.add_edge(Edge('SOCS-1', 'NFKB', -1))
service.add_edge(Edge('SOCS-1', 'MAL', -1))
```

Listing 4.2: Representation of MAL using pyrthomas

Once we have the BRN represented as a network, NetworkAnalyser can be used to generate state graph and analyse it.

NetworkAnalyser is the second module in pyrthomas. Using this module, we can generate state graphs using the qualitative modeling approach and check for stable and deadlock states. In listing 4.3 we first initialise network analyser and then generate all possible state graphs. Then we loop over all possible stategraphs to check cycles and deadlocks.

```
from pyrthomas.network import NetworkAnalyser
analyser = NetworkAnalyser(service.network)
state_graphs = analyser.get_possible_state_graphs()
for state_graph in state_graph:
    cycles = analyser.get_cycles(state_graph)
    deadlock_states = analyser.get_deadlock_states(state_graph)
```

Listing 4.3: Compute all possible state graphs using PyRThomas

4.4 Performance

We computed all possible stategraphs of BRNs from Dataset. For BRN of pseudomonas aeruginosa given in figure 4.1 we were able to compute all possible graphs in 17 seconds. It was that quick because the network is so small, it has just two entities and three interactions. For the celeberaial maleria associated biological regulatory network given in figure 4.2 we were able to compute all possible stategraphs using developed framework in 8 minutes 56 seconds. The TLR3 associated biological regulatory network of dengue virus has 4 entities and 8 interactions given in figure 4.3, using PyRThomas we were able to calculate all possible stategraphs in 2 minutes 16 seconds. The Hexosamine Biosynthetic Pathway (HBP) is more denser than the ones discussed above. It has 9 entities with 17 interaction we were able to calculate 100000 state graphs in 1 hour which is only 0.000001455% of the expected possible number (68719476736) of state graphs. For less complex systems like malaria and dengue, the computational time is as short as 8.56 and 2.16 minutes respectively. The execution time for mentioned BRN is given in figure 4.7 represented in minutes. As the number of nodes and edges in the network grows, there is an exponential growth in complexity since a node needs to be evaluated in connection with different combinations of other nodes that affect the state of the given node. This complexity is still better than the other tools available, since they do not provide support for all state graphs to be computed in the first place. This will be discussed more in the next section. In future, it is planned to have parallelism builtin support to be able to execute in a cluster of computers.

4.5 Web Interface

As part of this work we have also developed a web based graphical user interface, which can be easily deployed on a server to allow the users to access remotely and easily input, manipulate and analyse BRNs. This GUI provides an editor to create nodes, create edges, import and export graphs, and view them in different layouts. It also support generating state graphs with specific parameters and find deadlocks and cycles in stategraph. Subsequent figures 4.8, 4.9, 4.10, 4.11, 4.12, 4.13 show the interface for the tool. Although not as dynamic as the command-based, it still has more functionality than is available for free with respect to the other tools in the market. Both nodes

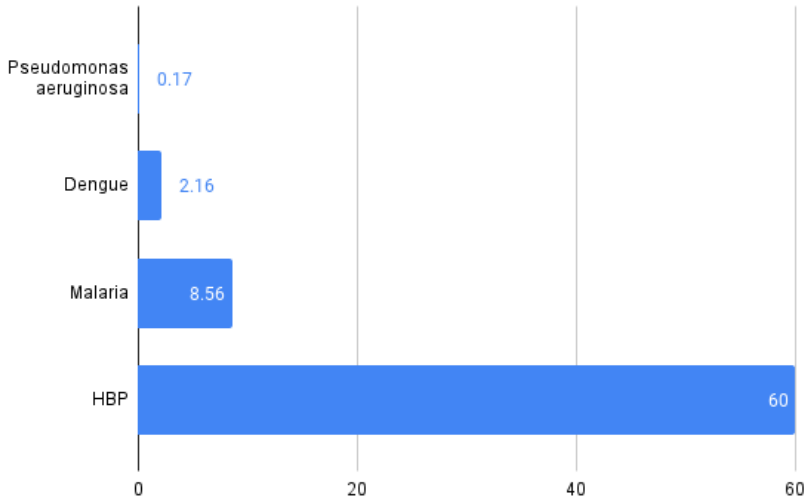

Figure 4.7: Execution time in minutes to generate all possible state graphs. For Pseudomonas aeruginosa we were able to calculate all 382 graphs in 0.17 min and for HBP we were able to process only 0.000001455% of possible state graphs.

and edges are input giving weight and directionality to each node and edge. In two-dimensional space the whole graph can be moved for ease of access to each node. this is important for looking at specific parts of the graph and their respective states. Figure 4.13 shows a state graph for the whole network, showing binary and mutual relationship between all the nodes with an epicenter in the middle. The René Thomas' framework can subsequently be run on the graph to model its behavior across time. These parameters are also also present in the interface.

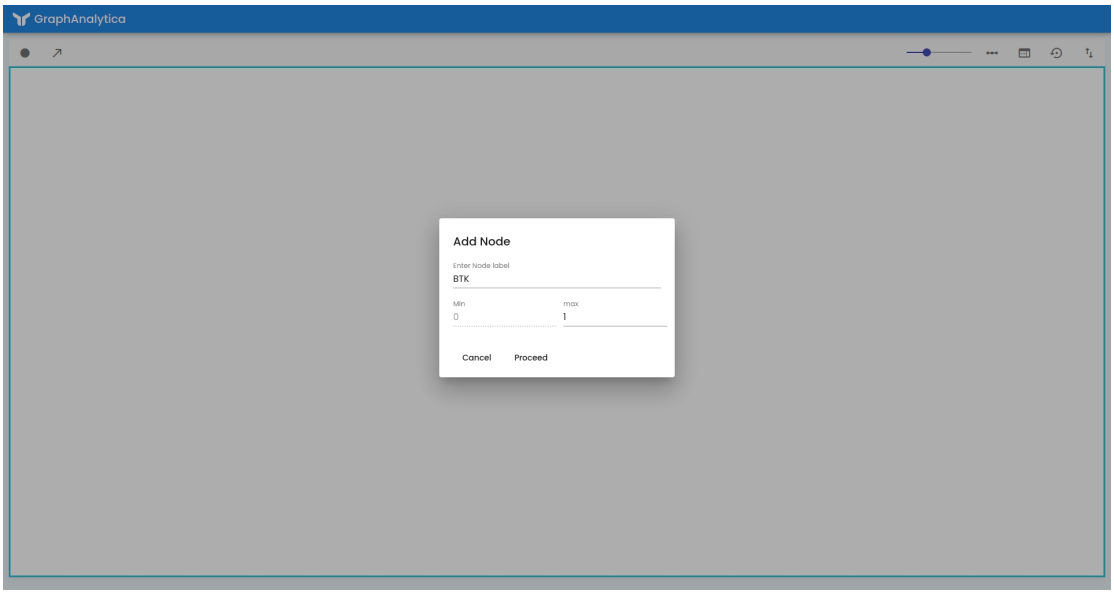

Figure 4.8: User can add node by clicking on the dot icon on top left and specify the name of the node and maximum threshold that a certain node can reach.

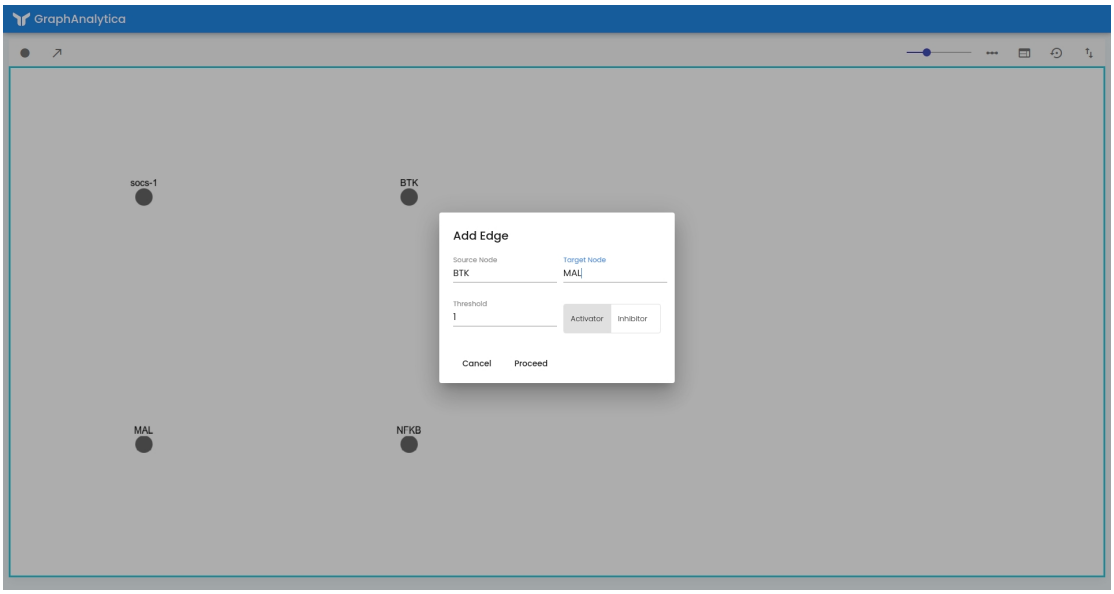

Figure 4.9: User can add edge for a network using the arrow button on top left. Pressing the button will ask to specify source node, target node, threshold and directionality of threshold (Activation/Inhibition).

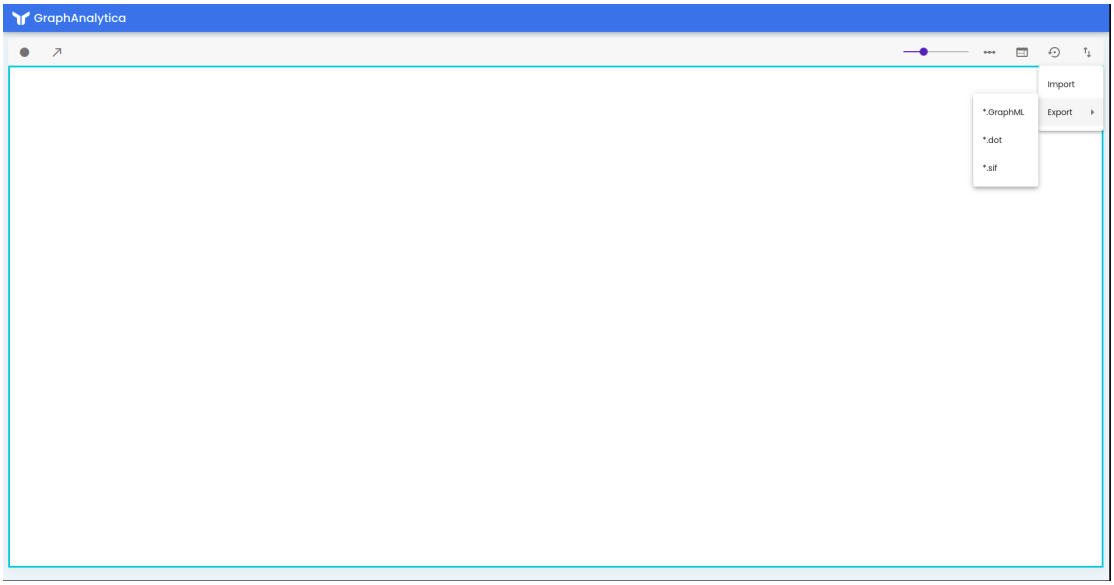

Figure 4.10: Import, Export menu: User can use import and export buttons on top right corner to save/restore the network

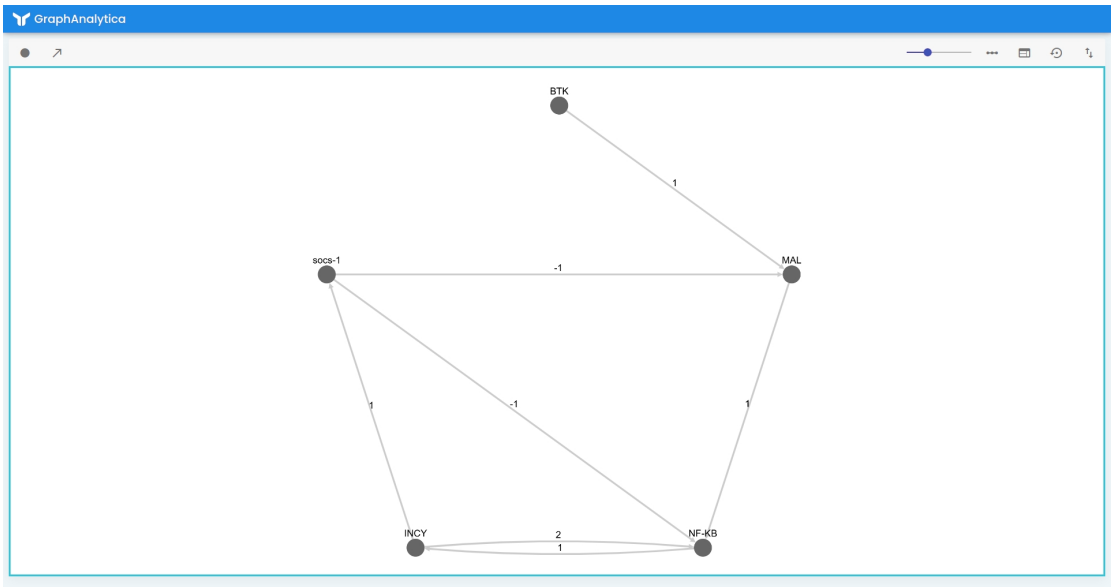

Figure 4.11: Biological regulatory network of Cerebral Malaria.

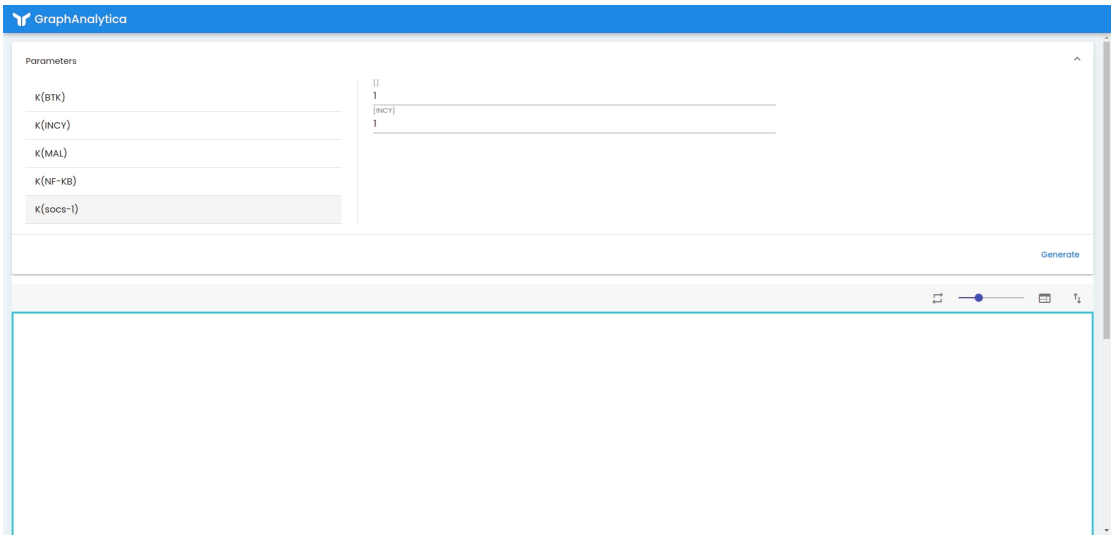

Figure 4.12: User can input parameter for each of the K state.

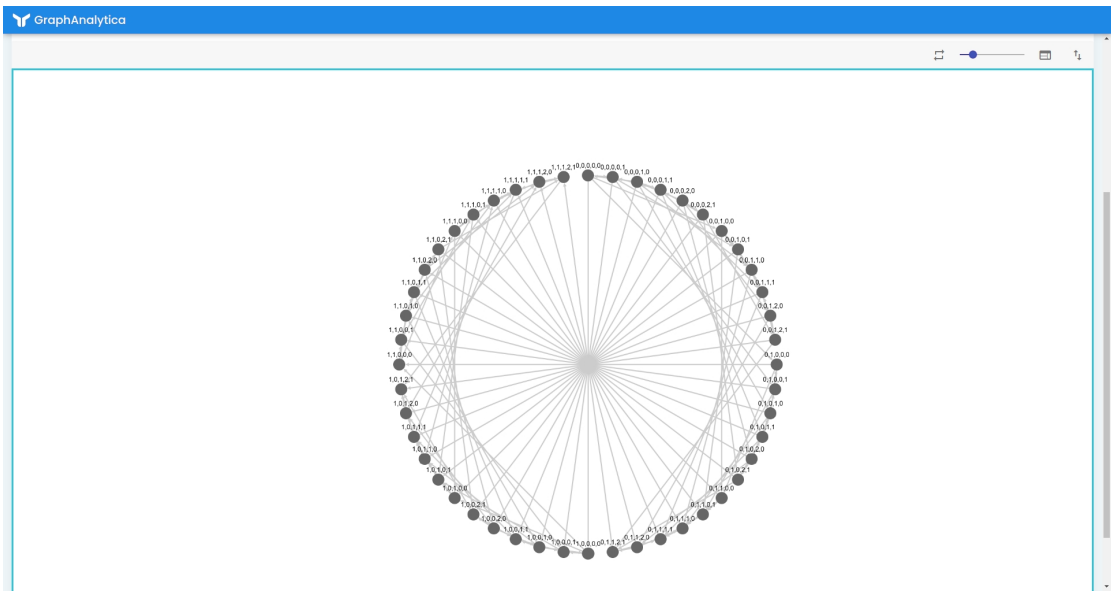

Figure 4.13: After parameters input pressing "Generate" creates a State graph of Cerebral malaria with given parameters. Shown in a circle layout.

Chapter 5

Discussion

PyRThomas is built considering the limitations faced in the usage of existing tools for modelling biological regulatory networks. We developed PyRThomas using python programming language to provide easy to use application programming interface, and a graphical user inteface to make the visualization of BRNs easier. Already available tools to analyse the biological regularity networks based on René Thomas' formalism are GenoTechE, GINsim and SMBioNet. The comparison of PyRThomas with existing tools is shown in table 5.1. For instance GenoTech which is a graphical tool, is not open source nor does it offer API to the user. Similarly, GINsim just like GenoTechE does not provide API or open-source code base but rather has both graphical and command-line based interface however, web Application is not available for GINsim. on contrary widely used application SMBioNet is open source but does not offer APIs. In addition it does not have any graphical user interface GUI, only command line facility is available, which often compromise its use by the researchers. Here we have developed an application PyRThomas to overcome the discussed limitations of existing tools. In contrast to all these tools, PyRThomas is an open-source application, that does not provides command line facility however the graphical interface with API and web application that allows quick prototyping is provided. Moreover, it also check for already available Thomas framework based tools for BRN modeling in the industry. Table 5.1 provided shows each tool with their respective strengths and limitations marked as Yes or No. As reported in chapter 4, PyRThomas was used to draw and interpret complex Biological Regulatory Network.

Although the our developed Python based Thomas framework can perform what we

Table 5.1: Comparison With Existing tools: PyRThomas is Open Source, has an API available, with a Web application which allows quick prototyping. As it provides GUI which is web base, it can be deployed to a remote server.

Tool	Open Source	API	GUI	Command Line	Web Application
GenoTechE	No	No	Yes	No	No
GinSim	No	No	Yes	Yes	No
SMBioNet	No	No	No	Yes	No
PyRThomas	Yes	Yes	Yes	No	Yes

intended, however some drawbacks are there. The major limitation is that at the moment it is only able to do the processing on a single compute core. Biological systems are generally complex in nature, the denser BRNs with large number of nodes and edges can have state graphs in billions, processing on a single core can be challenging for PyRThomas.

Chapter 6

Conclusion

The study of complex biological pathways, their abstraction into biological regulatory networks and analysis to extract useful information constitutes an important domain in systems biology. The qualitative modelling framework proposed by René Thomas is helpful in reducing the complexity of problem compared to quantitative methods. A number of tools such as GinSim have been developed that can be used to analyze biological networks. An important requirement however is the availability of open source libraries and packages that can be easily extended and integrated with the existing codes and packages. To come up with an application that can cater existing limitations of already available tools is need of the hour.

In this study, we developed a Python framework based on Reneé Thomas' qualitative modelling formalism. PyRThomas can be used to construct biological network by defining object of a network service. Afterwards entities like genes and proteins can be added in the form of nodes and elicit all possible interactions between them. Once the network is defined in PyThomas, Network Analyser can be invoked to run different algorithms such as building a dynamic stategraph, compute cyclic circuits and deadlock states etc. The state-graph can be imported in different formats for further interpretation. To check the performance of PyRThomas we utilized pseudomonas, dengue, malaria, and the Hexosamine Biosynthetic Pathway gene as case studies. Our application was robust enough to build and analyse these networks in relatively less time. In addition it provides different useful operations that include defining or importing a new network (using GraphML, dot or sif) formats, setting various layouts for existing network, and invoking different operations in a GUI mode. The GUI application is also open source

that can be used to extend the frontend to support new features. The availability of an application programming interface, comprising of various key functions enables the development of future applications in a modular manner.

Chapter 7

Recommendation

This research can be reproduced and extended in different directions. One of the main challenges underlying the analysis of complex biological networks is the computational complexity of the algorithms for computation of cycles and stable states. The challenge can be addressed by incorporating model reduction techniques into existing framework. Similarly by providing a support for parallel execution of some of the computational intensive procedures such as finding out stable states and cycles, the processing time for the large networks can be reduced. This service is available for SMBioNet but even that is provided as a web-server limiting its scalibility. In future PyRThomas can be modified to generate all possible state-graphs of estimated parameters for the user to read and interpret graphically at the same time.

In this study, we have also developed a frontend application that is provided with the existing framework. The aforementioned application can be used to write future applications that are GUI based. The current GUI supports most of the options/features provided by the API.

Appendices

Appendix A

Supplementary Content

A.1 MIT License

Pyrthomas - an open source biological network analysis tool based on the Rene-Thomas algorithm

Author: Talha Junaid

Institution: Research Center for Modelling and Simulation, NUST, Pakistan

License: MIT License

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

A.2 Implementation of NetoworkService in PyRThomas

```
# CODE STARTS HERE
import networkx as nx
from networkx.readwrite import json_graph

from .constants import pickle_key, graphml_file_name, dot_file_name, sif_file_name
from .cytoscape import read_sif, write_sif
from . import utils


class NetworkService:
    """Bild a network and add characteristics to it."""
    network: nx.DiGraph

    def __init__(self) -> None:
        try:
            self.network = nx.read_gpickle(pickle_key)
        except FileNotFoundError:
            self.network = nx.DiGraph()
            nx.write_gpickle(self.network, pickle_key)

    def add_node(self, node):
        """Add a node to the network."""
        self.network.add_node(node.id, min=node.min, max=node.max)
        self.persist_network()

    def add_edge(self, edge):
        """Add edge weights."""
```

```
        self.network.add_edge(edge.source, edge.target, weight=edge.weight)
        self.persist_network()

    def get_nodes(self):
        """Get total nodes built so far."""
        return list(self.network.nodes)

    def clear(self):
        """Collapse the whole network."""
        self.network.clear()
        self.persist_network()

    def get_edges(self):
        """Get edge weights."""
        return list(self.network.edges)

    def import_graphml(self, file):
        """Import network from a graphml file."""
        self.network = nx.read_graphml(file)
        self.persist_network()
        return json_graph.node_link_data(self.network)

    def import_dot(self, file):
        """Import network from a dot format file."""
        self.network = nx.drawing.nx_pydot.read_dot(file)
        self.persist_network()
        return json_graph.node_link_data(self.network)

    def import_sif(self, file):
        """Import network from sif file format."""
        self.network = read_sif(file)
        self.persist_network()
        return json_graph.node_link_data(self.network)
```

```
    def export_graphml(self):
        """Export to a graphml format."""
        file = open(graphml_file_name, "wb")
        nx.write_graphml(self.network, file, prettyprint=True)
        file.close()
        return file.name

    def export_dot(self):
        """Export network to a dot file format."""
        file = open(dot_file_name, "w")
        nx.drawing.nx_pydot.write_dot(self.network, file)
        file.close()
        return file.name

    def export_sif(self):
        """Export network in a sif file format."""
        file = open(sif_file_name, "w")
        write_sif(self.network, file)
        file.close()
        return file.name

    def persist_network(self):
        """Parse the whole network."""
        utils.persist_network(self.network, pickle_key)
```

A.3 Implementation of NetworkAnalyser in PyRThomas

```
import itertools
from typing import Tuple, Dict, List

from networkx import DiGraph, nx
from networkx.classes.reportviews import EdgeView, NodeView
```

```
from pyrthomas import utils


class NetworkAnalyser:
    """Analyze the network for given sets of properties."""

    @staticmethod
    def set_predecessor_combinations(network: DiGraph):
        """Add predecessor information to the network."""
        for node in network.nodes:
            predecessors = frozenset(nx.DiGraph.predecessors(network, node
                ))
            combinations = frozenset(utils.all_subsets(predecessors))
            nx.set_node_attributes(network, {node: combinations}, '
                predecessor_combinations')

    @staticmethod
    def get_state_space(graph: DiGraph) -> List[Dict[str, int]]:
        """Get the total state space of a network."""
        max_thresholds = [NetworkAnalyser.get_max_threshold(graph, node)
            for node in graph.nodes]
        state_space = tuple(itertools.product(*max_thresholds))
        return [dict(zip(graph.nodes, state)) for state in state_space]

    @staticmethod
    def get_max_threshold(graph: DiGraph, node: NodeView) -> Tuple[int]:
        """Get the maximum cut-off for the BRN and use it for analysis."""
        edges = graph.edges(node, data=True)
        max_weighted_edge = max(edges, key=lambda x: utils.get_weight(x,
            absolute=True))
        max_threshold = utils.get_weight(max_weighted_edge, True)
        return tuple(range(max_threshold + 1))
```

```
    @staticmethod
    def get_state_graph(network: DiGraph, parameters) -> DiGraph:
        """Get the total state graph for the network."""
        state_space = NetworkAnalyser.get_state_space(network)
        state_graph = nx.DiGraph()
        state_space_nodes = [utils.create_node_from_dict(state) for state
            in state_space]
        state_graph.add_nodes_from(state_space_nodes)

        resources = NetworkAnalyser.calculate_resources(network,
            state_space)

        k_states = NetworkAnalyser.calculate_k(resources, parameters)

        next_states = NetworkAnalyser.calculate_next_states(k_states)

        edges = NetworkAnalyser.generate_edges(next_states)
        state_graph.add_edges_from(edges)
        state_graph.graph['parameters'] = parameters
        return state_graph

    @staticmethod
    def generate_edges(next_states):
        """Generate edges for the network."""
        edges = list()
        for key, states in next_states.items():
            previous_value = dict(key)
            source = utils.create_node_from_dict(previous_value)
            for state in states:
                target = utils.create_node_from_dict(state)
                edges.append((source, target))
        return edges
```

```
    @staticmethod
    def calculate_next_states(k_states):
        """Using the Rene-Thomas formulation, calculate the next state of
           a given set of nodes."""
        for state_key, state in k_states.items():
            previous_entities = dict(state_key)
            new_val = list()
            temp_previous = previous_entities.copy()
            for entity_key, previous_entity in previous_entities.items():
                next_value = k_states[state_key][entity_key]

                if next_value > previous_entity:
                    previous_entities[entity_key] = previous_entity + 1
                elif next_value < previous_entity:
                    previous_entities[entity_key] = previous_entity - 1

                if not (previous_entities in new_val) and not
                   previous_entities == temp_previous:
                    new_val.append(previous_entities.copy())
                    previous_entities[entity_key] = temp_previous[
                        entity_key]
            k_states[state_key] = new_val
        return k_states

    @staticmethod
    def calculate_k(resources, parameters):
        """Calculate the k, rate constant for the biological network."""
        for resource_key, entities in resources.items():
            for entity_key, entity in entities.items():
                matched_interaction = filter(lambda x: sorted(x[0]) ==
                    sorted(entity),
                                             parameters[entity_key])
```

```
                first_interaction = next(matched_interaction)
                entities[entity_key] = first_interaction[1]
        return resources

    @staticmethod
    def get_cycles(network: DiGraph):
        """Get the total number of cycles."""
        return nx.simple_cycles(network)

    @staticmethod
    def get_deadlock_states(network: DiGraph):
        """Get nodes that go into deadlock, either because of auto-
            regulation or endpoint."""
        out_degree_iter = network.out_degree(network.nodes)
        return [node for node, out_degree in out_degree_iter if out_degree
             == 0]

    @staticmethod
    def calculate_resources(network: DiGraph, state_space: List[Dict[str,
        int]]):
        """Calculate available resources for the biological network."""
        resources = dict()
        for state in state_space:
            node = dict()
            for key in state:
                entity_resources = list()
                in_edges = network.in_edges(key, data=True)
                for edge in in_edges:
                    is_resource = NetworkAnalyser.is_resource_of_state(
                        state, edge)
                    if is_resource:
                        entity_resources.append(edge[0])
                entity_resources.sort()
```

```
                node[key] = entity_resources
            state_key = tuple(state.items())
            resources[state_key] = node
        return resources

    @staticmethod
    def get_required_parameters(network: DiGraph):
        """Given a network, find all the required parameters needed to be
            input."""
        nodes = network.nodes
        parameters = dict()
        for node in nodes:
            predecessors = frozenset(nx.DiGraph.predecessors(network, node
                ))
            required_interactions = [(interaction, None) for interaction
                in
                                     utils.all_subsets(predecessors)]
            parameters[node] = required_interactions

        return parameters

    @staticmethod
    def is_resource_of_state(state: dict, edge: EdgeView):
        """Get vailable resources in a given state of the graph."""
        weight = utils.get_weight(edge)
        is_positive = weight >= 0
        value = state[edge[0]]
        return (is_positive and value >= abs(weight)) or (not is_positive
            and value < abs(weight))

    @staticmethod
    def get_possible_parameters(network: DiGraph):
        """Get all the possible parameters in a given network graph."""
```

```
        combinations = dict()
        for node in network.nodes:
            combinations[node] = list()
            max_threshold = utils.get_max_weighted_edge_threashold(network
                , node)
            predecessors = frozenset(nx.DiGraph.predecessors(network, node
                ))
            required_interactions = list(utils.all_subsets(predecessors))
            for arrangement in itertools.product(range(0, max_threshold +
                1), repeat=len(required_interactions)):
                arranged_combination = list(zip(required_interactions,
                    arrangement))
                combinations[node].append(arranged_combination)
        return [dict(zip(combinations, v)) for v in itertools.product(*
            combinations.values())]

    @staticmethod
    def get_possible_state_graphs(network: DiGraph):
        """Get all the possible state graphs."""
        all_params = NetworkAnalyser.get_possible_parameters(network)
        for param in all_params:
            yield NetworkAnalyser.get_state_graph(network, param)
```

References

[1] E. H. Snoussi, "Qualitative dynamics of piecewise-linear differential equations: a discrete mapping approach," *Dynamics and stability of Systems*, vol. 4, no. 3-4, pp. 565–583, 1989.

[2] M. Kaufman and R. Thomas, "Model analysis of the bases of multistationarity in the humoral immune response," *Journal of theoretical biology*, vol. 129, no. 2, pp. 141–162, 1987.

[3] E. H. Snoussi and R. Thomas, "Logical identification of all steady states: the concept of feedback loop characteristic states," *Bulletin of Mathematical Biology*, vol. 55, no. 5, pp. 973–991, 1993.

[4] R. Thomas and M. Kaufman, "Multistationarity, the basis of cell differentiation and memory. i. structural conditions of multistationarity and other nontrivial behavior," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 11, no. 1, pp. 170–179, 2001.

[5] O. Cinquin and J. Demongeot, "Roles of positive and negative feedback in biological systems," *Comptes rendus biologies*, vol. 325, no. 11, pp. 1085–1095, 2002.

[6] L. Glass and S. A. Kauffman, "The logical analysis of continuous, non-linear biochemical control networks," *Journal of theoretical Biology*, vol. 39, no. 1, pp. 103–129, 1973.

[7] M. Suderman and M. Hallett, "Tools for visually exploring biological networks," *Bioinformatics*, vol. 23, no. 20, pp. 2651–2659, 2007.

[8] H. De Jong, "Modeling and simulation of genetic regulatory systems: a literature review," *Journal of computational biology*, vol. 9, no. 1, pp. 67–103, 2002.

[9] C. Chaouiya, A. Naldi, and D. Thieffry, "Logical modelling of gene regulatory networks with ginsim," in *Bacterial Molecular Networks*, pp. 463–479, Springer, 2012.

[10] M. Ryan and M. Huth, *Logic in Computer Science: Modelling and Reasoning about Systems*. Cambridge University Press, 2000.

[11] J. Ahmad, U. Niazi, S. Mansoor, U. Siddique, and J. Bibby, "Formal modeling and analysis of the mal-associated biological regulatory network: insight into cerebral malaria," *PloS one*, vol. 7, no. 3, p. e33532, 2012.

[12] U. Rauf, U. Siddique, J. Ahmad, and U. Niazi, "Formal modeling and analysis of biological regulatory networks using spin," in *2011 IEEE International Conference on Bioinformatics and Biomedicine*, pp. 304–308, IEEE, 2011.

[13] D. Thieffry and D. Romero, "The modularity of biological regulatory networks," *Biosystems*, vol. 50, no. 1, pp. 49–59, 1999.

[14] C. Chaouiya, E. Remy, and D. Thieffry, "Petri net modelling of biological regulatory networks," *Journal of Discrete Algorithms*, vol. 6, no. 2, pp. 165–177, 2008.

[15] P. T. Monteiro, A. T. Freitas, D. Ropers, R. Mateescu, and H. de Jong, "Modeling and formal verification of biological regulatory networks: an integrative approach," in *2010 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)*, pp. 812–813, IEEE, 2010.

[16] M. Gustafsson, M. Hornquist, and A. Lombardi, "Constructing and analyzing a large-scale gene-to-gene regulatory network-lasso-constrained inference and biological validation," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 2, no. 3, pp. 254–261, 2005.

[17] C. Chaouiya, E. Remy, P. Ruet, and D. Thieffry, "Qualitative modelling of genetic networks: From logical regulatory graphs to standard petri nets," in *International Conference on Application and Theory of Petri Nets*, pp. 137–156, Springer, 2004.

[18] E. Remy and P. Ruet, "From minimal signed circuits to the dynamics of boolean regulatory networks," *Bioinformatics*, vol. 24, no. 16, pp. i220–i226, 2008.

[19] H. Siebert and A. Bockmayr, "Temporal constraints in the logical analysis of regulatory networks," *Theoretical Computer Science*, vol. 391, no. 3, pp. 258–275, 2008.

[20] M. Chen and R. Hofestaedt, "Quantitative petri net model of gene regulated metabolic networks in the cell," *In silico biology*, vol. 3, no. 3, pp. 347–365, 2003.

[21] I. Nachman, A. Regev, and N. Friedman, "Inferring quantitative models of regulatory networks from expression data," *Bioinformatics*, vol. 20, no. suppl_1, pp. i248–i256, 2004.

[22] N. Le Novère, "Quantitative and logic modelling of molecular and gene networks," *Nature Reviews Genetics*, vol. 16, no. 3, p. 146, 2015.

[23] J. Bordon, M. Moškon, N. Zimic, and M. Mraz, "Fuzzy logic as a computational tool for quantitative modelling of biological systems with uncertain kinetic data," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 12, no. 5, pp. 1199–1205, 2015.

[24] S.-Q. Wang and H.-X. Li, "Quantitative construction of regulatory networks using multiple sources of knowlege," in *2011 International Conference on Machine Learning and Cybernetics*, vol. 1, pp. 91–96, IEEE, 2011.

[25] G. Bernot, F. Cassez, J.-P. Comet, F. Delaplace, C. Müller, and O. Roux, "Semantics of biological regulatory networks," *Electronic Notes in Theoretical Computer Science*, vol. 180, no. 3, pp. 3–14, 2007.

[26] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin, "Application of formal methods to biological regulatory networks: extending thomas' asynchronous logical approach with temporal logic," *Journal of theoretical biology*, vol. 229, no. 3, pp. 339–347, 2004.

[27] J. Ahmad, J. Ashraf, K. Asghar, and U. K. Niazi, "Qualitative modelling and analysis of the regulatory network of indoleamine 2, 3-dioxygenase in tumour immune escape," in *International Conference on Computer Networks and Information Technology*, pp. 41–45, IEEE, 2011.

[28] C. Chaouiya, "Qualitative modelling of biological regulatory networks combining a logical multi-valued formalism and petri nets," in *2008 9th International Workshop on Discrete Event Systems*, pp. 263–268, IEEE, 2008.

[29] S. Badaloni, B. Di Camillo, and F. Sambo, "Qualitative reasoning for biological network inference from systematic perturbation experiments," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 9, no. 5, pp. 1482–1491, 2012.

[30] A. Naldi, C. Hernandez, N. Levy, G. Stoll, P. T. Monteiro, C. Chaouiya, T. Helikar, A. Zinovyev, L. Calzone, S. Cohen-Boulakia, *et al.*, "The colomoto interactive notebook: accessible and reproducible computational analyses for qualitative biological networks," *Frontiers in physiology*, vol. 9, p. 680, 2018.

[31] O. Grumberg, E. Clarke, and D. Peled, "Model checking," 1999.

[32] A. Pnueli, "The temporal logic of programs," in *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pp. 46–57, IEEE, 1977.

[33] E. M. Clarke and E. A. Emerson, "Design and synthesis of synchronization skeletons using branching time temporal logic," in *Workshop on Logic of Programs*, pp. 52–71, Springer, 1981.

[34] M. T. Saeed, J. Ahmad, J. Baumbach, J. Pauling, A. Shafi, R. Z. Paracha, A. Hayat, and A. Ali, "Parameter estimation of qualitative biological regulatory networks on high performance computing hardware," *BMC systems biology*, vol. 12, no. 1, p. 146, 2018.

[35] L. Paulevé, "Reduction of qualitative models of biological networks for transient dynamics analysis," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 15, no. 4, pp. 1167–1179, 2018.

[36] R. Samaga and S. Klamt, "Modeling approaches for qualitative and semi-quantitative analysis of cellular signaling networks," *Cell communication and signaling*, vol. 11, no. 1, p. 43, 2013.

[37] A. Stéphanou and V. Volpert, "Hybrid modelling in biology: a classification review," *Mathematical Modelling of Natural Phenomena*, vol. 11, no. 1, pp. 37–48, 2016.

[38] J. Fromentin, D. Eveillard, and O. Roux, "Hybrid modeling of biological networks: mixing temporal and qualitative biological properties.," *BMC systems biology*, vol. 4, p. 79, 2010.

[39] N. Tenazinha and S. Vinga, "A survey on methods for modeling and analyzing integrated biological networks," *IEEE/ACM transactions on computational biology and bioinformatics (TCBB)*, vol. 8, no. 4, pp. 943–958, 2011.

[40] Z. Khalis, J.-P. Comet, A. Richard, and G. Bernot, "The smbionet method for discovering models of gene regulatory networks," *Genes, genomes and genomics*, vol. 3, no. 1, pp. 15–22, 2009.

[41] J. Ahmad, G. Bernot, J.-P. Comet, D. Lime, and O. Roux, "Hybrid modelling and dynamical analysis of gene regulatory networks with delays," *ComPlexUs*, vol. 3, no. 4, pp. 231–251, 2006.

[42] A. G. Gonzalez, A. Naldi, L. Sanchez, D. Thieffry, and C. Chaouiya, "Ginsim: a software suite for the qualitative modelling, simulation and analysis of regulatory networks," *Biosystems*, vol. 84, no. 2, pp. 91–100, 2006.

[43] K. Artavanis-Tsakonas, J. Tongren, and E. Riley, "The war between the malaria parasite and the immune system: immunity, immunoregulation and immunopathology," *Clinical & Experimental Immunology*, vol. 133, no. 2, pp. 145–152, 2003.

[44] C. C. Khor, S. J. Chapman, F. O. Vannberg, A. Dunne, C. Murphy, E. Y. Ling, A. J. Frodsham, A. J. Walley, O. Kyrieleis, A. Khan, *et al.*, "A mal functional variant is associated with protection against invasive pneumococcal disease, bacteremia, malaria and tuberculosis," *Nature genetics*, vol. 39, no. 4, pp. 523–528, 2007.

[45] Z. Liang, S. Wu, Y. Li, L. He, M. Wu, L. Jiang, L. Feng, P. Zhang, and X. Huang, "Activation of toll-like receptor 3 impairs the dengue virus serotype 2 replication through induction of ifn-β in cultured hepatoma cells," *PloS one*, vol. 6, no. 8, p. e23346, 2011.

[46] R. M. de Queiroz, I. A. Oliveira, B. Piva, F. Bouchuid Catão, B. da Costa Rodrigues, A. da Costa Pascoal, B. L. Diaz, A. R. Todeschini, M. B. Caarls, and W. B. Dias, "Hexosamine biosynthetic pathway and glycosylation regulate cell migration in melanoma cells," *Frontiers in oncology*, vol. 9, p. 116, 2019.

[47] L. V. Pham, J. L. Bryant, R. Mendez, J. Chen, A. T. Tamayo, Z. Y. Xu, K. H. Young, G. C. Manyam, D. Yang, L. J. Medeiros, *et al.*, "Targeting the hexosamine biosynthetic pathway and o-linked n-acetylglucosamine cycling for therapeutic and

imaging capabilities in diffuse large b-cell lymphoma," *Oncotarget*, vol. 7, no. 49, p. 80599, 2016.

[48] L. Li, M. Shao, P. Peng, C. Yang, S. Song, F. Duan, D. Jia, M. Zhang, J. Zhao, R. Zhao, *et al.*, "High expression of gfat1 predicts unfavorable prognosis in patients with hepatocellular carcinoma," *Oncotarget*, vol. 8, no. 12, p. 19205, 2017.

[49] A. Vasconcelos-Dos-Santos, H. Loponte, N. Mantuano, I. Oliveira, I. De Paula, L. Teixeira, J. De-Freitas-Junior, K. Gondim, N. Heise, R. Mohana-Borges, *et al.*, "Hyperglycemia exacerbates colon cancer malignancy through hexosamine biosynthetic pathway," *Oncogenesis*, vol. 6, no. 3, pp. e306–e306, 2017.

[50] H. M. Itkonen, S. Minner, I. J. Guldvik, M. J. Sandmann, M. C. Tsourlakis, V. Berge, A. Svindland, T. Schlomm, and I. G. Mills, "O-glcnac transferase integrates metabolic pathways to regulate the stability of c-myc in human prostate cancer cells," *Cancer research*, vol. 73, no. 16, pp. 5277–5287, 2013.

[51] T. Dong, X. Kang, Z. Liu, S. Zhao, W. Ma, Q. Xuan, H. Liu, Z. Wang, and Q. Zhang, "Altered glycometabolism affects both clinical features and prognosis of triple-negative and neoadjuvant chemotherapy-treated breast cancer," *Tumor Biology*, vol. 37, no. 6, pp. 8159–8168, 2016.

[52] C. Yang, P. Peng, L. Li, M. Shao, J. Zhao, L. Wang, F. Duan, S. Song, H. Wu, J. Zhang, *et al.*, "High expression of gfat1 predicts poor prognosis in patients with pancreatic cancer," *Scientific reports*, vol. 6, no. 1, pp. 1–9, 2016.

[53] F. Duan, D. Jia, J. Zhao, W. Wu, L. Min, S. Song, H. Wu, L. Wang, H. Wang, Y. Ruan, *et al.*, "Loss of gfat1 promotes epithelial-to-mesenchymal transition and predicts unfavorable prognosis in gastric cancer," *Oncotarget*, vol. 7, no. 25, p. 38427, 2016.

[54] A. Horvath, S. B. Pakala, P. Mudvari, S. D. N. Reddy, K. Ohshiro, S. Casimiro, R. Pires, S. A. Fuqua, M. Toi, L. Costa, *et al.*, "Novel insights into breast cancer genetic variance through rna sequencing," *Scientific reports*, vol. 3, no. 1, pp. 1–13, 2013.

[55] S. Barry, C. Chelala, K. Lines, M. Sunamura, A. Wang, F. M. Marelli-Berg, C. Brennan, N. R. Lemoine, and T. Crnogorac-Jurcevic, "S100p is a metastasis-associated

gene that facilitates transendothelial migration of pancreatic cancer cells," *Clinical & experimental metastasis*, vol. 30, no. 3, pp. 251–264, 2013.

[56] B. H. Junker, D. Koschützki, and F. Schreiber, "Exploration of biological network centralities with centibin," *BMC bioinformatics*, vol. 7, no. 1, pp. 1–7, 2006.

[57] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[58] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conference* (G. Varoquaux, T. Vaught, and J. Millman, eds.), (Pasadena, CA USA), pp. 11 – 15, 2008.

[59] N. Jain, A. Bhansali, and D. Mehta, "Angularjs: A modern mvc framework in javascript," *Journal of Global Research in Computer Science*, vol. 5, no. 12, pp. 17–23, 2014.

[60] M. Franz, C. T. Lopes, G. Huck, Y. Dong, O. Sumer, and G. D. Bader, "Cytoscape. js: a graph theory library for visualisation and analysis," *Bioinformatics*, vol. 32, no. 2, pp. 309–311, 2016.

[61] S. Peres and C. Jean-Paul, "Contribution of computational tree logic to biological regulatory networks: Example from pseudomonas aeruginosa," in *Computational Methods in Systems Biology* (C. Priami, ed.), (Berlin, Heidelberg), pp. 47–56, Springer Berlin Heidelberg, 2003.