A Quantitative Systems Biology and Mechanistic Model of Synthetic Lethality – Defining Regulatory Pathways of Targeted Cellular Death in a Cancer Cell Line

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Paul Aiyetan
Master of Science
Johns Hopkins University, 2011
Doctor of Medicine
University of Ibadan, 2005

Director: Dr. Iosif Vaisman, Professor
Department of Bioinformatics and Computational Biology

Summer Semester 2021
George Mason University
Fairfax, VA

Dedication

This is dedicated to my loving parents, Mr. Amos Aiyetan and Mrs. Mary Aiyetan. Particularly to my Dad who I do know would have loved to see this in person but called to be with the Lord. Dad, this is for you.

Acknowledgments

An unquantifiable thanks to my parents who from birth have helped me dream big dreams, nurtured the creativity in me, and expended much of their resources to ensure I had a great start. To my teachers — too many to count, for believing and bringing out the best in me, thank you. A notable mention of Dr. Mayowa Owolabi, Dr. Regina Oladokun, Prof. Adesola Ogunniyi, Prof. Olayemi Omotade, and Prof. Effiong Akang (of blessed memory), who saw and always believed in me and encouraged my further academic pursuit post-medical school. Many thanks to Dr. Luigi Marchionni, Dr. Sarah Wheelan, and Dr. Kristina Obom at the Johns Hopkins University School of Medicine and School of Arts and Sciences who believed my dreams and accepted to mentor me fresh from clinical Medicine. Creating a position in his then new lab, Dr. Marchionni offered me my first practical opportunity to appreciate the utility of the computational biology field in translational research and Medicine. Many thanks also to Dr. Hui Zhang, Dr. Zhen Zhang, Dr. Daniel Chan and Dr. Lori Sokoll, at the Johns Hopkins University School of Medicine's Center for Biomarker Discovery and Translation, Pathology department. Their mentorship through my Pathology postdoctoral fellowship training, primarily sponsored through the National Cancer Institutes Clinical Proteomics Tumor Analysis Consortium (NCI/CPTAC) and the National Heart, Lung and Blood Institutes Programs of Excellence in Glycosciences (NHLBI/PEG) research funds, at the Johns Hopkins University School of Medicine planted a seed of the work presented here. A special thanks to my thesis advisers, Dr. Iosif Vaisman, Dr. Dmitri Klimov and Dr. Saleet Jafri for accepting to mentor the work presented here and for the significant time spent out of their busy schedules to attend to this. Thanks to Dr. Quong, for encouraging me to take a closer look at Fuzzy logic. My acknowledgement would be incomplete without mentioning the contribution of the high performance computing resources at the Frederick National Laboratory for Cancer Research (FNLCR) and the Extreme Science and Engineering Discovery Environment (XSEDE) supported by the National Science Foundation at the Texas Advanced Computing Center (TACC) at The University of Texas at Austin - http://www.tacc.utexas.edu. Without these resources, the computational cost of the work presented here would have been too demanding to be undertaken. And to Oluwafisayomi, for being patient with me and almost always succeeding in making me smile, thank you.

Table of Contents

List of Tables

List of Figures

Abstract

A QUANTITATIVE SYSTEMS BIOLOGY AND MECHANISTIC MODEL OF SYNTHETIC LETHALITY – DEFINING REGULATORY PATHWAYS OF TARGETED CELLULAR DEATH IN A CANCER CELL LINE

Paul Aiyetan, PhD

George Mason University, 2021

Dissertation Director: Dr. Iosif Vaisman

With an overall lifetime risk of about 4.3% and 4.0%, in men and women respectively, colorectal cancer remains the third leading cause of cancer-related deaths in the United States. In persons aged 55 and below, its rate increased at 1% per year in the years 2008 to 2017 despite the steady decline associated with improved screening, early diagnosis and treatment in the general population. Besides standardized therapeutic regimen, many trials continue to evaluate the potential benefits of vorinostat, mostly in combination with other anti-neoplastic agents for its treatment. Vorinostat is an FDA approved anti-cancer drug known as suberoylanilide hydroxamic acid (SAHA). It is a histone deacylase (HDAC) inhibitor which acts through many mechanisms to cause cancer cell arrest and death. However, like many other anti-neoplastic agents, resistance and or failures have been observed. In the HCT116 colon cancer cell line xenograft model, exploiting potential lethal molecular interactions by additional gene knockouts restored vorinotat sensitivity. This phenomenon, known as synthetic lethality, offers a promise to selectively target cancer cells. Although without clearly delineated understanding of underlying molecular processes, it has been demonstrated as an effective cancer-killing mechanism. In this study, we aimed

to elucidate mechanistic interactions in multiple perturbations of identified synthetically lethal experiments, particularly in the vorinostat-resistant HCT116 (colon cancer xenograft model) cell line. Given that previous studies showed that knocking down GLI1, a downstream transcription factor involved in the Sonic Hedgehog pathway – an embryonal gene regulatory process, resulted in restoration of vorinostat sensitivity in the HCT116 colorectal cancer cell line, we hypothesized that vorinostat resistance is a result of upregulation of embryonal cellular differentiation processes; we hypothesized that elucidated regulatory mechanism would include crosstalks that regulate this biological process. We employed a knowledege-guided fuzzy logic regulatory inference method to elucidate mechanistic relationships. We validated inferred regulatory models in independent datasets. In addition, we evaluated the biomedical significance of key regulatory network genes in an independent clinically annotated dataset. We found no significant evidence that vorinostat resistance is due to an upregulation of embryonal gene regulatory pathways. Our observation rather support a topological rewiring of canonical oncogenic pathways around the PIK3CA, AKT1, RAS/BRAF etc. signaling pathways. Reasoning that significant genes in this regulatory network and pathways are likely implicated in the clinical course of colorectal cancer, we show that the identified key regulatory network genes' expression profile are able to predict short- to medium-term survival in colorectal cancer patients – providing a rationale and basis for prognostication and potentially effective combination of therapeutics that target these genes along with vorinostat in the treatment of colorectal cancer.

Chapter 1: Introduction

The quest for effective therapies for colorectal cancer, particular in younger patients with advanced disease has never been more imperative. With an overall lifetime risk of approximately 4.3% and 4.0%, in men and women respectively[1, 2], colorectal cancer is the second leading cause of cancer-related deaths in the United States[3]. In persons aged 50 and below, its rate increased at 2% per year in the years 2012 to 2016 despite the steady decline associated with improved screening, early diagnosis and treatment in the general population[2,3]. According to the center for disease control and prevention (CDC), in 2017, 141, 425 new cases of colorectal cancers were reported, and 52, 547 people died of it[4]. The CDC estimates that for every 100, 000 people, 37 new colorectal cancer cases are reported and 14 people died of this cancer[4].

Historically, risk factors have been classified as modifiable and non-modifiable factors[5]. Modifiable factors have included being overweight, a sedentary lifestyle, diet rich in red and processed meat, and sugars, smoking and alcohol consumption, while non-modifiable factors include increasing age, history of inflammatory bowel disease, polyps, family history of colorectal cancer, ethnicity, type II diabetes mellitus, and familial or inherited syndromes [5]. Although familial or hereditary factors account for only a third of colorectal cancer diagnoses, their molecular basis have enabled fundamental understanding of the etiopathogenesis of the disease. These include, lynch syndrome (hereditary non-polyposis colon cancer or HNPCC) which is primarily associated with defects in the *MLH1*, *MSH2* or the *MSH6* genes, and accounts for about 2% to 4% of all colorectal cancers, familial adenomatous polyposis coli (FAP) which accounts for 1% of colorectal cancers, Peutz-Jeghers syndrome (PJS), and MUTYH-associated polyposis (MAP). Associated with mutations in the APC gene, the

FAP-related colorectal cancer consists of three sub-types with almost specific clinical features. These include: the attenuated FAP, associated with fewer polyps and development of colorectal cancer at a later age than it is typical; the Gardner syndrome, associated with tumors of the soft tissues, bones and skin; and the Turcot syndrome, associated with an higher risk of colorectal cancer and a predisposition to developing medulloblastoma – a brain cancer. Usually diagnosed at a younger age, PJS is associated with mutations in the *STK11* (*LKB1*) gene while as its name implies, MAP is caused by mutations in the *MUTYH* gene[5]. These associated genetic defects are characteristically those of genes involved in tumor suppression and DNA repair mechanisms [6].

Besides standardized therapeutic regimen, many trials continue to evaluate the potential benefits of vorinostat, mostly in combination with other anti-neoplastic agents for its treatment[7–17]. Vorinostat, an FDA approved anti-cancer drug known as suberoylanilide hydroxamic acid (SAHA), a histone deacetylase (HDAC) inhibitor, through many mechanisms, causes cancer cell arrest and death[18]. First discovered on attempts to make more efficient hybrid polar compounds that induce the differentiation of transformed cells[19, 20] and initially approved by the FDA for the cutanous manifestation of T cell leukemia, vorinostat has since become a therapeutic candidate for many tumors[21–29]. This is due in part to the evolving understanding of the role of epigenetic and posttranslational modifications in the etiopathogenesis of transformed cells[30–33]. Altering many pathways and processes, vorinostat has been discovered to not only alter the modification state of histone proteins but many more essential proteins involved in the oncogenic and tumor suppression process. More specifically and among many other mode of action, vorinostat inhibits the removal of acetyl group from the ϵ-amino group of lysine residues of histone proteins by histone deacetylases (HDACs). Accumulation of acetyl group maintains chromatin in an expanded state, facilitating transcriptional activities of major regulatory genes[18, 30, 34–36]. However, like many other anti-neoplastic agents, toxicities, resistance and or failures have been observed[13, 37, 38].

In the HCT116 colon cancer cell line xenograft model, exploiting potential lethal molecular interactions by additional gene knockouts, Falkenberg and colleagues were able to restore vorinotat sensitivity[39, 40]. This phenomenon, known as synthetic lethality, offers a promise to selectively target cancer cells[41]. Although without clear delineated understanding of underlying molecular processes, many studies demonstrate synthetic lethality as an effective cancer-killing mechanism.

In this study, we aimed to elucidate regulatory interactions, in multiple perturbations of identified synthetically lethal experiments, particularly in the vorinostat-resistant HCT116 (colon cancer xenograft model) cell line. In addition to elucidating interactions, we aim to elucidate key interactions that potentially determine observed phenotypes. Given that previous studies[39, 40] showed that knocking down GLI1, a downstream transcription factor involved in the Sonic hedgehog (SHH) pathway [42–44] – an embryonal gene regulatory process, resulted in restoration of vorinostat sensitivity in the HCT116 colorectal cancer cell line, we hypothesized that vorinostat resistance is a result of uptick in embryonal gene regulatory programs. We also hypothesized that elucidated regulatory mechanism would include crosstalks that regulate this biological processes – embryonal gene regulatory programs. We employed a knowledege-guided fuzzy logic regulatory inference method to elucidate mechanistic relationships from multiple synthetic lethal pertubation experiments in the vorinostat-resistant colon cancer cell lines. We validated inferred regulatory models in independent experiment datasets. And, we evaluated the biomedical significance of key regulatory network genes in an independent clinically annotated dataset.

To model such molecular interactions, we supposed a fuzzy approach would mitigate known challenges of modeling biological systems with high-throughput data. These include – inconsistencies and inaccuracies associated with high-throughput characterization, challenges of dealing with noise, and those of a semi-quantitative data [45]. Similar to Boolean networks, fuzzy methods are simple and are fit to model imprecise and or highly complex

networks [46, 47]. And, as opposed to differential equation-based models, they are relatively less computationally expensive and less sensitive to imprecise measurements [46–48]. The fuzzy approach compensates for the inadequate dynamic resolution of a Boolean (or discrete) network, while simultaneously addressing the computational complexity of a continuous network [49].

Added advantages with respect to using the fuzzy logic for expression dataset include; 1) by dealing with trends and not absolute values, fuzzy logic inherently accounts for noise in the data. 2) In contrast to other automated decision making algorithms, such as neural networks or polynomial fits, algorithms in fuzzy logic are presented in the same language used in day-to-day conversations. Therefore, a fuzzy logic is more easily understood and can be extrapolated in predictable ways. And, 3) fuzzy logic approaches can be scaled to include an unlimited number of components [50].

1.1 Significance and Rationale

1.1.1 Increasing National and International cancer burden

The need for a deeper understanding and approaches to combating cancer has never been more apparent. In the joint annual report of the American Cancer Society (ACS), the Centers for Disease Control and Prevention (CDC), the National Cancer Institute (NCI), and North American Association of Central Cancer Registries (NAACCR), the reported cancer incidence rates between the periods $2011-2015$ reduced by $2.1\%(95\%$ confidence interval [CI] $=$ $-2.6\%$to $-1.6\%)$ per year in males and were stable in females [51]. However, in spite of decreasing incidence rates, particularly among major cancers, the CDC in its estimates of cancer incidence, reports that the actual number of cases diagnosed each year had increased [52] and said to still increase. This picture represents that of a growing US population, more so among the older age group, given that the risk of being diagnosed with cancer generally increases with age. With increased life-expectancy and an aging population, it is

likely that this trend would continue into the immediate future [53–55]. Weir et al's CDC study predicts that between 2010 and 2020, total incident cases would increase by $> 20\%$ to approximately 1.9 million cases diagnosed each year. In more specific terms, a 24.1% to > 1 million cases in men and by 20.6% to $> 900,000$ annual cases in women. On a global scale, there were 17 million new cases of cancer worldwide in 2018, according to Cancer Research UK [56]. By 2040, according to the American Cancer Society, the global burden is expected to grow to 27.5 million new cancer cases and 16.3 million cancer deaths. This is also expected due to the growth and aging of the population. With increasing prevalence of risk factors such as unhealthy diet, physical inactivity, obesity, smoking and others, the future cancer burden will probably be considerably larger [57].

1.1.2 Rapidly evolving and deeper molecular profiling

From the days of Sanger sequencing methods, through those of next generation sequencing approaches at the wake of the current millennium, to contemporary times of much more sophisticated third and fourth generation sequencing machines, the amount of information per sequencing run has greatly increased at an exponential rate that far outpaces classical approaches to deriving meaningful sense from the data [58–61]. A typical Sanger sequencing, initially developed for small sized RNA molecules (about 75 - 120 base pairs) but extended to the DNA macromolecule, generated an output of a few hundred bases per sequencing run. Given advances in engineering including nanoscale miniaturization, increased parallelization of sequencing reactions with finer chemistries and molecular biology advances, the total number of sequence reads and consequently bases generated per run has tremendously increased. At the higher end of the spectrum, an Illumina HiSeq 2000 or 2500, which employs reversible and fluorescently labeled terminators in identifying sequence nucleotides has a capacity to generate up to 3 billion sequence reads and about 600 Gigabytes (Gb) bases per run. Trading off higher throughput for a significantly shortened runtime of about 4 hours, Life Technologies Ion torrent, which utilizes detectable change in pH level (proton release) in identifying sequence nucleotides has a capacity to identify about

4 million sequence reads and approximately 2Gb of bases per run. At the single molecule resolution scale lies the PacBio SMRT (single molecule real-time) and the Oxford nanopore-based sequencing technologies. The PacBio SMRT sequencer employs fluorescently labeled phospholinked nucleotides and highly efficient optical systems that can detect the incorporation of one fluorescently labeled nucleotide. Running over two days, this has the capacity to generate about 0.8 million reads and approximately 5 Gb bases per run. Better than any other sequencing technology, the nanopore-based technology provide ultra-long reads ($104 - 106$ bases) in addition to requiring lower starting material input.

In addition to higher throughput genetic interaction experiments utilizing RNAi, CRISPR-Cas9, and similar technologies mentioned earlier, deeper mass-spectrometry based proteomic, post-translational modifications (acetylation, ubiquitination, glucosylation, phosphorylation, etc) and metabolomic profiling, are providing real-time quantitative measures of subcellular macromolecules more than there ever had been.

These increasing ubiquitous and available datasets are providing a relatively rich starting material for deeper exploration – providing needed resources for system level integration and exploration of molecular interactions and regulations.

1.1.3 Paucity of mechanistic models explaining synthetic lethality

Since the completion of the human genome project, a large number of collaborative and institutionally supported large scale characterization programs have been embarked upon. Primarily supported by the National Cancer Institute (NCI) and the National Human Genome Research Institute (NHGRI), these notably include, the HapMap project [62–66], the Encyclopedia of DNA Elements (ENCODE) project [67–75], the Cancer Genome Atlas (TCGA) [76], the 1000 Genome Project [77–79] and more recently the Clinical Proteomic Tumor Analysis Consortium (CPTAC) [80]. Still somewhat in line with the NHGRI strategic plan for the period $1990 - 2003$ [81], these have essentially focused on establishing a complete

understanding of the structure of the human genome and products thereof. Although attempts at connecting the genome to biology and health is anticipated with respect to the CPTAC project, reported publication findings so far been predominantly generated using correlative models. A correlative model simply relates one quantity to another [82].

Despite its inarguably better rational to therapeutic target discovery and design, most of these quantitative cancer studies and many others, including those relating to synthetic lethality, have essentially been correlative in nature. It is without a doubt that appropriate mechanistic models stand to provide deeper insights into the regulation and control processes involved in biological processes.

As previously mentioned, synthetic lethality is no doubt a highly effective means of selectively killing cancer cells. Thus a careful and mechanistic elucidation of involved processes initiating, and resulting in this phenomenon, would not only improve our understanding but also provide possible means of utilizing such for therapeutics design.

An explanatory or mechanistic model, such as a Fuzzy logic model of regulation, would relate observations or outcomes of an experiment to biological processes and mechanisms that drive the phenomenon [82]. The Fuzzy approach is anticipated to provide a more rigorous treatment and an approach to derive more mechanistic information from large scale profile experiment data.

1.2 Novelty

The dissertation study would be the first to attempt to infer potential mechanistic models relating to synthetic lethality using the fuzzy logic approach, particularly in eukaryotes. This study would be a first to suggest direct molecular processes that converge on observed phenotype of neoplastic cell death in synthetic lethality. The dissertation work introduces the multi-staged hyper-parallel approach to address the computational time complexity that limits exhaustive searches at higher order regulatory models. Also, as a derivative work, the

platform-independent, integrated implementation of the Fuzzy logic Inference System (FIS) for biological data utilizing the URC (union rule configuration), packaged with a dynamic simulation and results post-processing modules, is also presumed to be a first.

1.3 Specific Aims

1.3.1 To extrapolate, using a fuzzy logic approach, regulators of cellular death in synthetic lethality

Here, we proposed to implement Sokhansanj et al's [49] scalable linear variant fuzzy logic approach, modeled after the union rule configuration (URC), developed by Combs and Andrew [83] to improve the robustness and generalization of the fuzzy model applied to expression data.

Along with the above, we proposed

1. Optimize the computational time complexity of the fuzzy inference approach.
2. Extract molecular interactions and regulations in the vorinostat-resistant colon cancer cell line model

1.3.2 To validate inferred regulators of cellular death in an independent dataset

On a similarly profiled independent dataset, we proposed to validate an optimally performing regulatory model, by comparing changes in network-inferring dataset to that observed in the independent dataset.

1.3.3 To investigate biomedical and clinical significance of major regulatory features in real-life biological data

We reason that significant regulatory network genes are likely implicated in the clinical course of colorectal cancer disease. Therefore, we aimed to evaluate the implication of the expression profile of these key genes on colorectal cancer patient survival.

Chapter 2: Background

2.1 Synthetic Lethality – An Overview

Synthetic lethality offers a promise to differentially target neoplastic cells. Since identified and proposed as a safer cancer killing mechanism [84], a handful molecular targeted therapies have employed it as an alternate and effective antineoplastic approach. Synthetic lethality is the phenomenon where the absence of the product of two genes selectively cause cellular death but individual deletion or absence of one of such does not. In other words, two genes are described as 'synthetically lethal' if mutations in either gene alone is compatible with viability but simultaneous mutation of both causes cellular death [85] [41] (Figure 2.1). Such identified and employed antineoplastic mechanism include; those of PARP inhibition in BRCA mutant cancers, topoisomerase II inhibition (etoposide) in pRB (RB) mutant cancers, HSP90 inhibition (17AAG) in BRAF mutant and EGFR mutant cancers, proteosomal inhibition (Bortezomib) in blood cancers, particularly multiple myeloma, mTOR inhibition in mutant PTEN (-/-) cells, and more recently ROS1 inhibition in CDH1 (E-cadherin) defective breast cancers [86–96]. These, among many others have been attributed to synthetic lethal or 'sickening' interactions. And, the effects of these have been associated with mechanisms like DNA damage, loss of cell-cycle checkpoints, oncogene addiction, and genetic streamlining.

More recently the concept of synthetic lethality or sickening has expanded to include other phenomena such as *synthetic dosage lethality* and *conditional synthetic lethality*. Beyond the loss-of-function or reduction-of-function paradigm, synthetic dosage lethality describes synthetic lethality in an alternate way - overexpression or underexpression of a member gene i.e. a genetic interaction whereby an underexpression of gene A combined

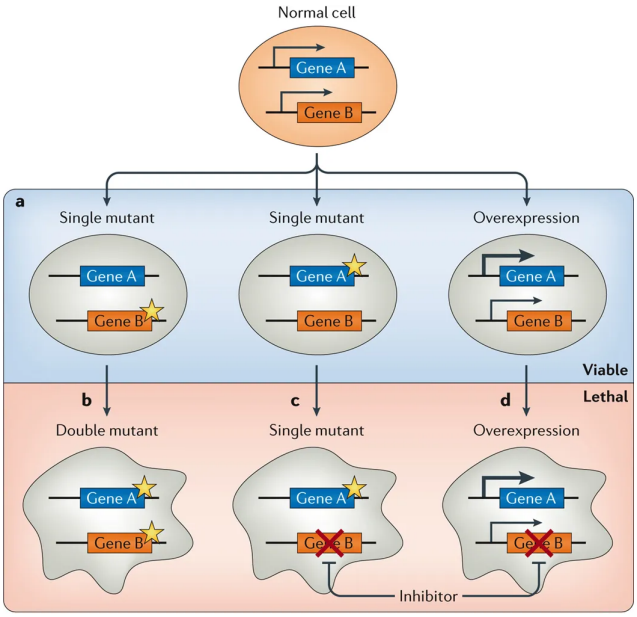


Figure 2.1: An illustration of synthetic lethality (O'Neil et al 2017). Two genes are described as synthetically lethal when simultaneous mutations or disruption of both genes function lead to cellular death (b, c, and d). This however does not occur when only one of the pair of genes function is disrupted or mutated. The loss or the inhibition of either of the protein products of gene A or B alone or the overexpression of gene A is viable (part a). Mutation (part b) or pharmacological inhibition (part c) of the protein product of gene B in cells with a mutation (parts b,c) or overexpression (part d) of gene A results in synthetic lethality. The thicker arrow denotes increased expression. The star shape denotes a mutation. The red crosses denote pharmacological inhibition. Viable cells are depicted as ovals, and inviable cells are depicted as random shapes

with an overexpression of gene B kills the cell [97,98]. Conditional lethality, also referred to as *context-specific* or *private synthetic lethality*, addresses interactions only observed in certain situations. Situations such as the cell's metabolic state, the cellular microenvironment, exposure to therapeutic agents, and the cell genetic background.

2.1.1 Screening Approaches

In spite of differences in phylogenetic relationships and oftentimes absence of homologous genes in eukaryotic counterparts, the search for synthetic lethal interactions has majorly been carried out in model organisms, particularly the yeast (Saccharomyces cerevisiae),

the fruit fly (Drosophila melanogaster) and the nematode worm (Caenorhabditis elegans). Compounded by the enormous possibilities of digenic or more interactions in varied contexts (such as hypoxic, radiation, chemotherapeutic, and metabolic states), the search for synthetic lethality naturally lent itself to high throughput methods to interrogate relevant interactions. Employed high-throughput screening methods have included; chemical screens, utilizing isogenic cells to identify compounds that selectively kill cancer cells as a result of synthetic interactions [99–108], and genetic screens using both forward and reverse approaches [109–117]. Genetic screens have historically involved the use of interfering ribonucleic acids (RNAi) or similar macromolecules, but more recently have been predominated by the use of the CRISPR-Cas9 (Clustered Regularly Interspaced Short Palindromic Repeats - Caspase 9) technology which targets the DNA instead [118–126].

Genetic screens

First introduced by Tong et al (2001), Synthetic Genetic Arrays (SGA) have been pivotal to high-throughput study of synthetic lethality. It is an automated method that combines arrays of genes with robotic manipulations for high-throughput construction of haploid yeast double mutants and identification of genetic interactions [127–130]. Arrayed genes are typically of either non-essential deletion mutants, or conditional alleles of essential genes.

Similar to SGA are the dSLAM (diploid-based synthetic lethal analysis by microarray) [131], the GIM (Genetic Interaction Mapping) [132, 133], and the epistatic miniarrays (E-MAPs) [134] methods. Developed by Pan X et al, dSLAM associates molecular 'barcodes' (TAGs) with knocked-out genes to facilitate quantitative profiling. In its truest sense, it is a collection of methods that include the SGA coupled onto a single platform. dSLAM extends the approach previously described by Ooi et al 2003 [135, 136] called SLAM (synthetic lethality by microarray). The often unpredictable nature of haploid mutants, in addition to potential genetic impurity necessitated a microarray-based TAG readouts. An effective selection of pure haploid strains combined with molecular tags is thought to provide a more

comprehensive, sensitive, accurate and fast functional characterization.

It would not be an overstatement that RNAi further paved the way for investigating synthetic lethal interactions in higher order organisms, particularly eukaryotes. Prior to the use of interfering RNAs to investigate 'epistatic interactions', synthetic lethality studies were constrained to lower organisms, particularly C. elegans and insights gained from such studies were marginally translatable to humans, as a sizeable number of genes do not have human homologs, as previously mentioned. In recent times, screening types have included; the negative selection ("drop-out"), positive selection ("drug resistance") and the transcriptional activator/repressor screens ("CRISPRa/i") [137].

Quantitative screens

To complement large scale biological or biochemical screens for synthetically lethal interactions, the need for computational approaches cannot be overemphasized. Methods employed have broadly consisted among others: 1) mutual exclusivity analyses 2) the hybrid approach and 3) the extensively data driven method, DAISY (data mining synthetic lethality identification pipeline). Mutual exclusivity analyses approaches are statistical methods that search for mutually exclusive mutations in genomic datasets or databases of known alterations. The discovery that such approaches have a shortfall of being bias toward more highly occurring mutations is being addressed in improvements in its algorithms [138, 139]. Hybrid approaches attempt to improve the mutual exclusivity analyses by not only using copy number variation information but also associated information on cell signaling, mutation and gene expression [138, 140, 141]. The extensive data driven approach, DAISY developed by Jerby-Arnon et al combined three main data sources: i) cell lines and clinical sample data on somatic mutations and copy number alterations, ii) essential genes (required for proliferation or viability in the context of a single cell line or tumor type) profile from RNAi screens and iii) cell line derived gene expression data [142, 143]

2.1.2 Phenotype Measurements

Fundamental to most genetic interaction experiment or screen, particularly the genetic screening approach, is the phenotypic readout. From single numerical values to multidimensional images derived from automated microscopy, RNAi and CRISPR-based approaches present a rich assortment of phenotypic information as a result of genetic manipulation [144]. Numeric values may represent those of specific reporters measuring particular biological reactions or measures of model organisms' biological viability. To improve the generalization, accuracy and ease of interpretation of screen results, multiple reporters have been used. Typical markers used for reporting include, the GFP (green fluorescent protein), luciferase enzyme activity readout, caspase enzyme activity readout, cell titre fluorophore (CTF) readout etc. In addition to measuring viability, reporters may inform the activity of particular pathways or pathway components [145–147]. Though historically, luminescence readouts have served as surrogates for phenotype measurement, more contemporary approaches have proposed alternate phenotype measures. For example, the ATARiS (Analytic Technique for Assessment of RNAi by Similarity) described a "gene-level phenotype" value, derived from considering observed patterns in RNAi data across multiple samples to enrich for RNAi reagents whose phenotypic effects relate to suppression of their intended targets [148]. Methods related to this approach have included the 'redundant siRNA activity' (RSA) [149] and the 'strictly standardized mean difference' (SSMD) [150,151], for which in each sample, observed phenotypes for all genes screened are considered simultaneously (RSA) or separately (SSMD).

2.2 Fuzzy Logic and Fuzzy Sets

The Fuzzy logic is based on partial or imprecise classification of entities. It attempts to describe an entity across multiple classifications. It ascribes a degree of membership for each possible class an entity may be classified. In some other words, entities that constitute a class are specified to a level of truth or degree of membership. For example, a pink ball

may be described as being partly red and partly white. The description of such class the ball belongs can be said to be fuzzy as it may well be said to be red and white to some respectively specified degrees. Building on prior work by Bellman, another colleague and himself, Zadeh formalized the 'fuzzy set' concept within a mathematical framework. In the seminal publication, the authors introduced the 'fuzzy set' as a framework for pattern recognition, whose purview prior to then had been to classify patterns into a finite number of categories. However, in their work, they described a fuzzy set to extend the concept of membership in a set to situations in which there are many, or possibly a continuum of grades of membership [152, 153]. Given a universe of objects, U, a subset (fuzzy set) or class of objects A can be described by applying a function (membership function, f) on a random selection of objects, X to derive a numeric value in the range $[0, 1]$. An element, x_i in X can be said to belong to class A if derived value is greater than zero and the nearer the value of $f_A(x)$ to unity, the higher the 'grade of membership' of x in A. When A is a set in the ordinary sense of the term, its membership function can take only two values 0 and 1 [154]. In which case respective elements xiin X are either not of or are of the class A.

$$f_A(x) = \begin{cases} 1 \text{ if } x \in A \\ 0 \text{ if } x \notin A \end{cases}$$

For a fuzzy set, different functions (membership functions) on A, f_A can be considered. This is typically subjective and context dependent [155].

2.2.1 Fuzzy Set Operations

The notion of a set lends to fuzzy set, ordinary set operations – union, intersection, and complement from the Naive and classic set theory [156] [157]. Given ordinary sets A and B,

$$A \cup B = \{x : x \in A \text{ or } x \in B\}$$

$$A \cap B = \{x : x \in A \text{ and } x \in B\}$$

$$A' = \{x \in U : x \notin A\}$$

The union of set A and B is a set whose respective element x is in set A or set B. An intersection is a set whose respective element is an element in both sets A and B. A complement of a set A, described as A', is a set whose elements are contained in the universal set U but not in set A.

Before describing a projection of the above onto fuzzy sets, the set theory constructs, inclusion and equality as they apply to fuzzy sets are worth a first mention.

$A \subset B$, if $\forall x$, $f_A(x) \leq f_B(x)$

A fuzzy set A is a subset of (or said to be included in) fuzzy set B if for all elements x in A, the value of the membership function applied to x in A is less than or equal to the value thereon in B [158]. And for set equality,

$A = B$ if and only if $\forall x$, $f_A(x) = f_B(x)$

A fuzzy set A is equal to fuzzy set B if and only if for all elements x in A, the value of the membership function applied to x in A is equal to the value thereon in B [158]

Now, unions, intersections, and complements in terms of membership functions and

fuzzy sets,

$$f_{A\cup B}(x) = \max\{f_A(x), f_B(x)\} = f_A(x) \vee f_B(x)$$

$$f_{A\cap B}(x) = \min\{f_A(x), f_B(x)\} = f_A(x) \wedge f_B(x)$$

$$f_A(x) = 1 - f_A(x)$$

Alternately written as:

$$(A \vee B)(x) = \max\{A(x), B(x)\} = A(x) \vee B(x)$$

$$(A \wedge B)(x) = \min\{A(x), B(x)\} = A(x) \wedge B(x)$$

$$A'(x) = 1 - A(x)$$

The union of two fuzzy sets A and B with respective membership functions $f_A(x)$ and $f_B(x)$ is a fuzzy set C, written as $C = A \cup B$, whose membership function is related to those of A and B by $f_{A\cup B}(x) = max\{f_A(x), f_B(x)\}$ - abbreviated as $f_A(x) \vee f_B(x)$. More intuitively the union of A and B is the smallest fuzzy set containing both A and B. If D is any fuzzy set containing both A and B, then it also contains the union of A and B [153].

The intersection of two fuzzy sets A and B with respective membership functions $f_A(x)$ and $f_B(x)$ is a fuzzy set C, written as $C = A \cap B$, whose membership function is related to those of A and B by $f_{A\cap B}(x) = \min\{f_A(x), f_B(x)\}$ - abbreviated as $f_A(x) \wedge f_B(x)$. It is the largest fuzzy set which is contained in both A and B [153].

The description of a fuzzy set complement is slightly different from that of an ordinary

set because, there isn't the notion of an element 'belonging to' a particular set or universal. Rather, individual elements of the reference set speak of the degree of membership to the specified set within the range [0,1].

To address other ways of combining fuzzy sets, Zadeh (Zadeh 1965) also described the following fuzzy operations; algebraic product, algebraic sum, absolute difference, and convex combinations. The algebraic product of A and B is defined as:

$f_{AB} = f_A f_B$ Which translates to:

$AB \subset A \cap B$

Although only meaningful when the condition $f_A(x) + f_B(x) \leq 1$, the algebraic sum is specified as :

$f_{A+B} = f_A + f_B$

Denoted by $|A - B|$ the absolute difference is defined as:

$f_{|A-B|} = |f_A - f_B|$

In a vector space, given a finite number of vectors $\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_n$, their convex combination is a vector of the form $\lambda_1\vec{v}_1 + \lambda_2\vec{v}_2 + \cdots + \lambda_n\vec{v}_n$ where the real numbers satisfy $\lambda_i \in [0, 1]$ and $\lambda_1 + \lambda_2 + \cdots + \lambda_n = 1$ [159,160]. Zadeh's (Zadeh 1965) original specification of a convex combination operation can be said to describe two fuzzy sets.

2.2.2 Logical Reasonings with Fuzzy Sets

In its simplest description, fuzzy logic is the use of fuzzy sets in the representation and manipulation of vague information for the purpose of making decisions or taking actions [155]. It is a form of many-valued logic in which the truth values of variables may be any real number between 0 and 1 both inclusive, and employed to handle the concept of partial truth, where the truth value may range between completely true and completely false [161].

By attributing truth a degree between absolute false and definite true, fuzzy logic refines but contrasts Boolean logic, where the truth values of variables may only be the integer values 0 or 1. It is a formal adaptation of prior studied many-valued logic popularized by Jan Łukasiewicz in the early part of the 20th century.

Historical study of logic and inference

The systematic study of the forms of arguments and inference dates back to Aristotle. And historically, the semantic principle of bivalence states that every declarative sentence expressing a proposition (of a theory under inspection) has exactly one truth value which is either true or false. Such logic is called a two-valued logic [162][163]. An extension of the classical two-valued logic to more than two values is called n-valued logic. Notable among these are the three-valued (such as Łukasiewicz's and Kleene's), the finite-valued (finitely-many valued) with more than three values, and the infinite-valued (infinitely-many-valued), e.g. fuzzy logic and probability logic. Many other specific examples do abound in the literature. Many of these involve products of works of the immediate past century. These include the Priest's logic of paradox, the Bochvar's internal three-valued logic, Belnap logic, Gödel logics, Product logic, Post logics, Rose logics, among others [164–166].

Classical logic, fuzzy logic and probability

Classical logic and the many variants of the aforementioned many-valued logics permit conclusions which are either true, false, indeterminate, unknown, etc. But quite often time a perfect classification is not quite possible as with previously described pink ball example. More fuzzy or vague would it be described when the color the said ball falls within a tone of shade. Possible descriptions can be mapped onto a spectrum. Fuzzy logic mathematically models these vagueness or spectrum of possible description by employing degrees of truth. It is important to mention here that, though similar to probability in terms of range of value of between $[0, 1]$, fuzzy logic is not probability [155, 167]. Though a persisting discourse, the forms of uncertainty addressed in both are different. It is argued that Zadeh developed

the fuzzy logic concepts as a response to the lack of probability theory for jointly modelling uncertainty and vagueness [168]. As questions of degrees of belief in mutually-exclusive set membership in probability theory can be represented as certain cases of non-mutually-exclusive graded membership in fuzzy theory, Bart Kosko argues that probability theory is a subtheory of fuzzy logic [169]. In his treatise, he derived the Bayes' theorem from the concept of fuzzy subsethood as a proof. Fuzzy logic extends classical logic to address uncertainty outside of classical logic and situations not amenable to probability theory.

Fuzzy linguistic variables, truth values and approximate reasoning

To enable the expression of rules and facts, fuzzy logic, many a times, utilizes non-numeric values [170]. These are referred to as linguistic variables. Linguistic variables are variables whose values are words or sentences in a natural or artificial language [171]. Examples include age, whose values may include: young, not young, very young, quite young, old, not very old, not very young, etc. Another example is temperature. Its values may include: very hot, slightly hot, very warm, slightly warm, cold etc. The word qualifiers - very, slightly, quite etc are described as hedges. A collection of all values of a linguistic variable is referred to as term-set. The numeric values associated with a linguistics variable, when such exist, are called base variables e.g. $1, 2, 3\ldots$ for the variable age. Linguistic variables are structured - having two associated rules: i) syntactic rule and ii) semantic rule. Syntactic rules govern how the values in the term-set are generated, while semantic rules provide how to compute meaning of any linguistic value [171]. In formal terms, a linguistic variable X is specified by a quintuple:

$(X, T(X), U, G, M)$

Where X is the name of the variable;
$T(X)$ is the term-set of X;
U is a universe of discourse;
G is a syntactic rule which generates the terms in $T(X)$; and

M is a semantic rule which associates with each term x in $T(X)$ its meaning, $M(x)$

$M(x)$ denotes a possibility distribution in U. The meaning of x is defined by a membership function or, equivalently, a possibility distribution function [172].

Treating truth as a linguistic variable whose truth-values form a term-set brings linguistic variable concepts into the realms of approximate reasoning - a fuzzy logic which may well be a better approximation to the logic involved in human decision processes than the classical two-valued logic [171][173]. E.g.

$T(Truth) = \text{true} + \text{not true} + \text{very true} + \text{completely true} + \text{more or less true} + \text{fairly true} + \text{essentially true} + \cdots + \text{false} + \text{very false} + \text{neither true nor false} + \cdots$

The underlying base variable, in this case is the interval $[0, 1]$, and the meaning of a primary term such as *true* is identified with a fuzzy restriction on the values of the base variable. The compatibility function (also called membership function as earlier described) is a mapping from the unit interval to itself [171, 174].

2.3 Fuzzy Logic in Regulatory Inference

Exploring interactions between biological macromolecules, and elucidating causal relationships between these and biological phenomena remains the purview of high-throughput biomedical research. With both parallel and converging advances in quantitative and computational approaches, the capacity to explore small- to genome-scale or systems-wide interactions continue to tend to be within reach. Within the past three decades of research, methods of arriving at biological inference have spanned techniques such as correlation coefficients, information theory, regression analyses, network analyses and many more. With

respect to network analyses, employed approaches have included Boolean, Bayesian (including naive Bayes), artificial neural networks, ODE (ordinary differential equation) based methods, and fuzzy logic approaches.

2.3.1 Significance

A fuzzy approach is thought to mitigate known challenges of modeling biological systems. These include inconsistencies and inaccuracies associated with high-throughput characterizations. There are challenges of dealing with noise and those of dealing with a semi-quantitative data [175]. Similar to Boolean networks, fuzzy methods are simple and are fit to model imprecise and or highly complex networks [176, 177]. But, as opposed to differential equation based models, they are less computationally expensive and less sensitive to imprecise measurements [176,177][178]. The fuzzy approach compensates for the inadequate dynamic resolution of a Boolean (or discrete) network, while simultaneously addressing the computational complexity of a continuous network [179].

Three advantages exist with respect to using the fuzzy logic for expression dataset; First, an inherent account for noise in the data. Fuzzy logic deals with trends, not absolute values. Second, in contrast to other automated decision making algorithms, such as neural networks or polynomial fits, algorithms in fuzzy logic are presented in the same language used in day-to-day conversation. Therefore, a fuzzy logic is more easily understood and can be extrapolated in predictable ways. Lastly, fuzzy logic approaches are computationally efficient, and can be scaled to include an unlimited number of components [180].

2.4 The Fuzzy Logic Inference and Control System

A general fuzzy logic based modeling and control system entails three major steps (Figure 3.3):

1. Fuzzification

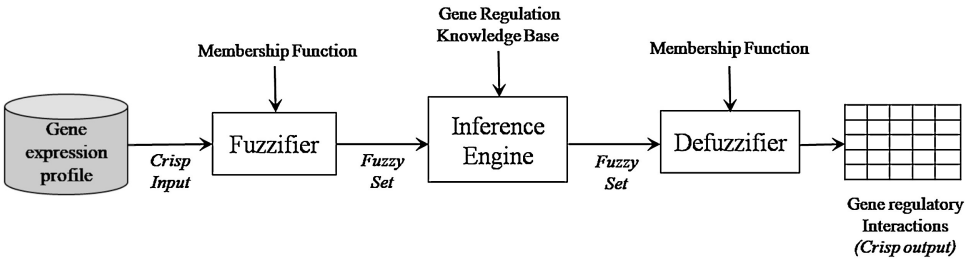


Figure 2.2: A generic pipeline of fuzzy logic model of GRN inference(Raza 2019)

2. Rule evaluation, and

3. Defuzzification

[181].

2.4.1 Fuzzification

Considering expression as a linguistic variable and applying defined membership functions on observed continuous numerical expression data, the fuzzification step derives qualitative values. It is a mapping of non-fuzzy inputs to fuzzy linguistic terms [181]. To make data fuzzification easier, a normalization technique may be applied to scale values to within a preferred range [179, 181, 182].

2.4.2 Rule evaluation

Driven by an inference engine, constructed rules in the form of "IF-THEN" are used to evaluate input variables and draw inference on the outputs. The fuzzy set operations (AND, OR, or NOT) earlier described are used to evaluate the fuzzy rules. The evaluation step attempts to make an expert judgment of collective liguistic terms. It attempts to find a solution to an evaluation of concurrent state of existense of liguistic description of states. Several methods can be used to aggregate results into a definitive output. These include the maximum, bounded sum or normalized sum methods [181].

2.4.3 Defuzzification

The defuzzification step produces a quantifiable expression result or value given the input sets, the fuzzy rules, and membership functions. Defuzzification technically interpretes the membership degrees of the fuzzy sets into a specific decision or real value. The defuzzification step attempts to report a corresponding continuous numerical variable from a fuzzy state liguistic variable. Several approaches to defuzzify abound. The most common of these is the center of gravity approach - it computes the center of gravity of the area under the membership function [183]. Where X is an ordinary non-void set, a mapping A from X into the unit interval [0, 1] is the a fuzzy set on X, the value $A(x)$ of A in $x \in X$ is the degree of membership, the center of gravity defuzzification is given by [183]:

$$COG(A) = \frac{\sum_{x_{min}}^{x_{max}} x.A(x)}{\sum_{x_{min}}^{x_{max}} A(x)}$$

Other methods that are variants of the COG method include the basic defuzzification distributions (BADD) [184], mean of maxima (MeOM), indexed center of gravity (ICOG) among others [183].

2.4.4 Classical Fuzzy Logic in Regulatory Network

Woolf and Wang (2000) presented one of the first applications of fuzzy logic to elucidate regulatory networks. Describing gene expression levels in linguistic terms of three possible states – low, medium, and high, they sort to find interacting gene triplets modeled as targets (T), activators (A), and repressors (R). Membership functions were employed to characterize expression levels as LOW, MEDIUM, and HIGH. With these, quantitative set of rules were used to model regulatory networks. A sample predefined rule takes the form of "if A is LOW and R is HIGH, then T is LOW (Table 2.1). On each possible triplet, the expression linguistics were tested against the rules presented in the Table.

Table 2.1: Woolf and Wang's rule table

	HIGH	MED	LOW
LOW	LOW	LOW	MED
MED	LOW	MED	HIGH
HIGH	MED	HIGH	HIGH

In other words, their method entailed fuzzifying the expression data; creating and comparing gene triplets (activator-repressor-target) to generate a prediction value for the target (T) at points where the predicted values of A and R overlap i.e rule evaluation; and defuzzification to derive crisp values of target predictions and triplet screening. Screening entailed comparing target predictions against observed expression values across biological experiments.

2.4.5 Improving Performance

Almost immediately apparent is the computational complexity that is associated with Woolf and Wang's approach – scaling in exponential time on the order $O(n^3)$ [180], where n is the number of interacting molecules. This quickly limits the number of interacting molecules that can be modeled to only three, i.e. two inputs and one output. Without improvement, the algorithm may only model simple regulation patterns and unable to scale well to more complex models whose implementation time would be on the scale of years instead of hours [185]. Extending preliminary works of Reynolds [186], by modifying the data preprocessing step, Ressom and others did improve Woolf and Wang's approach by up to 50% [187]. Reduction in computation time was achieved by introducing clustering as a preprocessing step. This reduced the number of gene combinations to be analyzed without any effect on the results [187][181]. With added focus on the preprocessing step, Ram et al also extended Woolf and Wang's work. By grouping genes having similar changes in expression profile over available intervals in the microarray data, they eliminated redundant computation performed by the model [188]. Ram et al and Ressom et al's approaches appear somewhat similar because they both essentially are in search of a minimal set of network features using

clustering-like methods.

The Union Rule Configuration

The problem of the exponential growth in the number of rules as inputs, compromising performance, associated with the intersection-rule configuration obtained in conventional fuzzy inference methodology of Woolf and Wang's and others ([180, 187, 188] was partly addressed by Combs and Andrews [189]. In their paper, Combs and Andrews had proposed an alternative rule configuration called the union-rule configuration (URC), together with a corresponding rule matrix called the union-rule matrix (URM) [190], to model the entire problem space without incurring any combinatorial penalty. Having first demonstrated the utility of the URC to qualitatively model the lac operon of E. coli [191][191,192], Sokhansaj et al extended the URC approach to model the yeast cell cycle from a time series expression data. In addition, their elucidated model was capable of qualitatively predicting data from another time series experiment [179].

Analyzing the Fuzzy Logic Algorithm

Computational-time Complexity

As earlier mentioned, the impact of the computation algorithm employed can significantly affect the utility of the fuzzy logic approach to elucidate regulatory network. The classical fuzzy logic triplet model of Woolf and Wang is reported to run on the order $O(n^3)$. Where n is the number of interacting molecules. This is a very conservative estimate. It accounts for only the number of fuzzy rule evaluations performed for a specific combination (activator-repressor-target) of a particular set of triplet. It does not account for those of other combinations nor does it account for all other possible triplets. These can have a combinatorial explosion-like growth function that may quickly become significant in comparison to that observed with the rules evaluated with increasing n. Employing the union-rule configuration (URC), Sokhansanj et al were able to reduce the complexity of Woolf and Wang's

solution from $O(m^{N^N})$ to $O(m^N)$. Where N is the number of (input) genes regulating an output gene and m is the number of possible rules describing the effect of each single input gene on an output gene. The number of possible rules for each gene-gene interaction (m) is given by n^n, where n is the number of fuzzy sets that describe the state of a variable [179]. Similarly, this is a very conservative estimate. It accounts for only the number of fuzzy rule evaluations performed for a specific combination of a particular set of inputs (regulators) and output genes. It does not account for those of other combinations of input genes nor does it account for all other possible combinations of inputs (regulators) and output genes which may similarly exhibit a combinatorial explosion-like growth function.

2.5 Feature Selection

On a one hand is the cost of learning a regulatory model using the fuzzy logic approach, but on another hand is the curse of dimensionality, that plagues the low sample to feature ratios characteristic of biological experiments. Although optimized search algorithms, such as mentioned above, may mitigate cost, poorly selected or less optimal set of features are set to undermine the efficiency of the learned model. Feature selection seeks to find a middle ground where cost is minimized without or minimal loss of the learning benefits. To provide a basis and justification for the subset of features selected for the fuzzy logic model in this dissertation study, here in this chapter is highlighted feature selection with particular respect to methods employed for regulatory network inference. The methods for estimating feature relevance and subset search methods with associated criterion functions are highlighted. And, a few classical algorithms implementing these methods are summarized.

2.5.1 Feature Selection for Regulatory Networks

Although similar, feature selection for regulatory network inference differs from classical feature selection. The types of problems aimed at addressing by classical and regulatory network feature selection may greatly differ. Classical feature selection [193–197] approaches aim to identify the optimal set of features with which a training algorithm can best predict or

correctly identify a class given the set of features with not-previously-seen feature attributes. When it involves data labels, it is referred to as supervised [198] and unsupervised when otherwise [199, 200] [201]. Those involving partial data labels are referred to as semi-supervised. Also for class prediction problems, the argument of feature redundancy [202] comes to the fore of the selection process. This may not necessarily be the case, with respect to selecting features for regulatory networks since features that appear redundant may imply a co-regulatory (direct or indirect regulatory) mechanism in a network of interest.

Regulatory networks can range from small networks of a few features to very large networks of hundreds or thousands of regulatory elements, all of which may play a significant role in terms of the larger systems functions. However, with larger sized networks comes the problem of 'overfitting', as more features need to be modeled with relatively few available samples. And thus, the need for reduction in dimension. The very high dimension coupled with low sample size and the potential noise in measured experiments present a limitation for regulatory network inference methods [203]. As with non-regulatory-network-related dimensionality reduction methods, two methods exist: feature extraction [204] and feature selection. Because of the preservation of feature properties needed to make biological interpretation of inferred model meaningful, feature selection is a preferred method for dimensionality reduction.

Feature selection for regulatory networks consists of estimating relevance of features, and based on estimated feature relevance, one or more combinations of filtering and or some search mechanisms are employed to determine an optimal set of features; it is composed of two parts – a search algorithm and a criterion function [205]. Search algorithms can on the one hand be exhaustive, returning the best feature subspace, and by so doing be computationally expensive. On another hand, the search algorithm can be suboptimal – trading off bits of quality of derived feature subspace for modest computational cost [205]. As with unsupervised feature selection methods, many existing feature evaluation criteria can be unified under a common formulation, where the relevance of features is quantified by

measuring their capability in preserving sample similarity specified by a predefined property [206].

Fundamentally, irrespective of it being supervised, unsupervised, semi-supervised or for biological regulatory network inference, feature selection aims to improve the cost of the learning process from the data at hand.

2.5.2 Feature Relevance Estimation

An initial step in selecting features is an estimation of the relevance of individual features. Features are said to be relevant if their values vary systematically with category membership [207]. Without labels, categorical memberships or attributable classes, relevant features maybe such as have the inherent property to distinguish between potential class memberships or represented states in a given dataset, independent of other features or together with a few others. Inherent properties often considered have included, variance, range, and other measures of dispersion. With labels, categorical memberships or attributable classes and assuming all features and labels are boolean without noise, Almuallim and Dietterich described a feature X_i as relevant$C$ if it appears in every Boolean formula that represents C, and irrelevant otherwise [208] [209, 210]. In probabilistic terms, a feature X_i is said to be relevant if there exists some x_i and y for which $p(X_i = x_i) > 0$ such that

$$p(Y = y|X_i = x_i) \neq p(Y = y)$$

That is, X_i is relevant if knowing its value can change the estimates for Y, in other words, if Y is conditionally dependent on X_i.
Also a feature X_i is relevant if there exists some x_i, y and s_i, for which $p(X_i = x_i) > 0$ such that

$$p(Y = y, S_i = s_i | X_i = x_i) \neq p(Y = y, S_i = s_i)$$

That is, X_i is relevant if the probability of the label (given all features) can change when the knowledge about the value of X_i is removed. And also a feature may be regarded as relevant if there exists some x_i, y, and s_i, for which $p(X_i = x_i, S_i = s_i) > 0$ such that

$$p(Y = y | X_i = x_i, S_i = s_i) \neq p(Y = y, S_i = s_i)$$

In the same work, John et al showed that these definitions may give unexpected results and thus introduced the concept of the degree of relevance of features, described as strong or weak. A strongly relevant feature is such that cannot be removed without the loss of prediction accuracy, one such that $p(Y = y | X_i = x_i, S_i = s_i) \neq p(Y = y, S_i = s_i)$, while a feature X_i is weakly relevant if it is not strongly relevant and there exist a subset of features S'_i of S_i for which there exists some x_i, y, and s'_i with $p(X_i = x_i, S'_i, s'_i) > 0)$ such that

$$p(Y = y | X_i = x_i, S'_i = s'_i) \neq p(Y = y | S'_i = s'_i)$$

With respect to regulatory networks, it appears more rational to evaluate features as subsets rather than individually. Individual evaluation, also known as feature weighting or ranking [197][195], assesses individual features and assigns them weights according to their degrees of relevance [202]. Subset evaluations for regulatory networks employ criterion functions to assess relevance. Lopes et al 2011 described three types of criterion functions: 1) the correlation based 2) the Bayesian error estimation based and 3) the information theory based [203]. The correlation based approach assesses the pairwise relationships between genes, functional modules, and clusters [211]. Bayesian error estimation based criterion functions on the other hand, evaluate the estimated errors present in the joint

probability distribution of a target gene given its candidate predictor genes [212–214]. While the Bayesian approach is able to detect N-to-1 relationship among features, the correlation approach only evaluate 1-to-1 relationships - it does not take into account multivariate relationships, i.e., the expression of a given target being regulated by a set of two or more genes with multivariate interaction [203]. Combining benefits of both the correlation and Bayesian based approaches, the Information theory based criterion function detects 1-to1 as well as N-to-1 relationships [215–219]. It relies on the uniformity of the conditional probability distributions of the target given the candidate predictors with higher uniformity implying higher entropy and thus smaller mutual information [203]. It would be acceptable to add tree-based approaches to the list. Modeled after decision-trees [220], tree-based approaches are able to tease non-linear 1-to-1 and N-to-1 relationships, and implicitly select for relevant features [221, 222].

Mutual Information Theory Approach

Mutual information is intricately linked to the concept of entropy. Entropy spans many physical science fields [223, 224]. However, in information theory, it is the expected amount of information held by a random variable. With respect to two random variables, mutual information is the measure of mutual dependence. It can be described as how much information about the second variable (in bits, also called 'shannons' unit) is learned or appreciable from the knowledge of the first variable. The mutual dependence may be quantified by calculating the average amount in the uncertainty or probability on some variable v_i given the knowledge of that of the other variable v_k, and vice-versa [225]. Mutual information measures the information that two variables share i.e. how much knowing one of these variables reduces uncertainty about the other. Expressed in terms of the joint distribution of X and Y relative to the joint distribution of X and Y.

$I(X;Y) = 0$, iff X and Y are independent random variables i.e.

$$
\begin{aligned}
P_{(X,Y)}(x,y) &= P_X(x) \cdot P_Y(y) \\
&\implies \log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x)\,p_Y(y)}\right) \\
&= \log 1 \\
&= 0
\end{aligned}
$$

With Jensen's inequality, mutual information $I(X;Y)$ is proven to be non-negative [226] and traditionally it is expressed as

$$
I(X;Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\frac{p(x,y)}{p(x)\,p(y)}
$$

In terms of two random features v_i and vk [225, 227]

$$
\begin{aligned}
I(v_i;v_k) &= \sum_{v_i}\sum_{v_k} P(v_i,v_k)\ln\left(\frac{P(v_i,v_k)}{P(v_i)P(v_k)}\right) \\
&= S(v_i) - S(v_i|v_k)
\end{aligned}
$$

where

$$
S(v_i) = -\sum_{v_i} P(v_i)\ln P(v_i)
$$

and

$$S(v_i|v_k) = -\sum_{v_k} P(v_k) \sum_{v_i} P(v_i|v_k) \ln P(v_i|v_k)$$

are the Boltzmann-Gibbs entropy of the gene v_i and its conditional entropy on the gene v_k, also known as the Shannon entropy and its conditional entropy, respectively.

Probabilistic Theory Approach

Berrara et al described genetic network in probabilistic terms as a finite dynamical system, discrete in time and composed by a finite number of states. Within this system, each transcript is represented by a variable. The composition of all variables form a vector considered the system state. Each vector component has an associated transition function which calculates its next value from the previous state of other genes [203, 219]. Given $R = 0, 1$ in binary systems, and $R = -1, 0, 1$ in three levels systems, the transition function ϕ, for a gene network of n genes, is a function from R^n to R^n. For a finite dynamic system,

$$x[t+1] = \phi(x[t]), x[t] \in R^n \text{ for every } t > 0, \text{ and the transition function } \phi \text{ is the same}$$

When for each state $x[t]$, the next state $\phi(x[t])$ is a realization of a random vector, implying that ϕ is a stochastic function, the dynamical system is referred to as a stochastic process governed by a Markov Chain. A probabilistic genetic network (PGN) treated as a Markov chain $(\pi_{Y|X}, \pi_0)$, is characterized by a transition matrix $\pi_{Y|X}$ of conditional probabilities between states, whose elements are denoted $p_{y|x}$, and an initial condition random vector of states π_0. Such network assumes [219]

1. $\pi_{Y|X}$ is homogeneous, i.e. $p_{y|x}$ is not a function of t.

2. $p_{y|x} > 0$, for every pair of states $x, y \in R^n$.

3. $\pi_{Y|X}$ is conditionally independent, i.e. for every pair of states $x, y \in R^n$

$$p_{y|x} = \prod_{i=1}^{n} p(y_i|x)$$

4. $\pi_{Y|X}$ is almost deterministic, i.e. for every state $x \in R^n$, there exists a single state, $y \in R^n$ such that $p_{y|x} \approx 1$.

As noted by Margolin et al (2006), temporal gene expression data are difficult to obtain for higher eukaryotes, and cellular populations harvested from different individuals generally capture random steady states of the underlying biochemical dynamics. Therefore, the use of methods that infer temporal associations and thus plausible causal relationships, such as the above are often precluded[215][228].

Correlation-based Approach

The correlation based approach for find the most relevant feature simply computes the pair wise correlation among all features and selects features with the highest correlation values. This may be combined with other approaches to provide additional evidences to support relevances of features. Using a correlation approach, and by virtue of coexpression networks, Stuart et al identified functionally relevant expression modules across species. After computing correlation between every pair of genes (meta-genes), an estimate of the probability of observing the gene-gene correlations by chance was also computed. The metagenes were used to represent genes with varying orthologous names across species. It facilitated a consistent naming of specific genes across all species datasets considered. After correcting for multiple-testing, pairs of metagenes with significant adjusted p-value were connected [211].

Tree-based Approach

Implicit in the tree construction process is the estimation of the relevance of ultimate members of the chosen tree. Exemplified in Huynh-Thu et al (2010), tree based approaches re-define the network inference problem as a feature selection problem. For steady state experiments, it assumes that the expression of each gene is a function of the expression of the other genes in the network, plus some random noise. If x_k^{-j} is the vector representing the expression values of all genes in experiment k except gene j i.e.

$$x_k^j = (x_k^1, \cdots, x_k^{j-1}, x_k^{j+1}, \cdots, x_k^p)^T$$

then

$$x_k^j = f_j(x_k^{-j}) + \epsilon_k, \forall k$$

Where ϵ_k is the random noise with zero mean [221]. Decomposing the problem with p genes into p different subproblems for which the regulators of each gene is sought from an expression profile data, i.e. the subset of genes whose expressions are predictive of the expression of the target gene. Tree-based ensembles come handy in elucidating the relationship between features and the target gene because they i) makes no assumption of the nature of the functions f_j ii) can work with interacting features and non-linear relationships iii) work well with a large number of features, iv) computationally fast and scalable, and v) require no parameter estimation i.e. parameter-free [221].

2.5.3 Feature Subset Search Methods

Pudil et al, described the feature selection problem as one that detects an optimal feature subset based on a selected measure - evaluated by a suitable criterion function [229]. In a "bottom-up" or "top-down" manner, a feature subset, X_d, is selected by adding or removing from a set of features till a desired subset size of d cardinality is attained. Pudil and

colleagues introduce the the floating search methods to address 'nesting effect' methods associated with Marill and Green's sequential backward selection (SBS) method [230] and Whitney's sequential forward selection (SFS)[231]

Sequential Forward Search (SFS) and Sequential Backward Search (SBS)

These differ by virtue of their starting number of features added to a combination of optimal features determined by a chosen criterion function. On the one hand, the SFS starts with an empty set of features and progressively adds a new feature to the set based on a determined 'best' feature. The best feature is that which together with already selected features offers the best predictive ability as determined by the criterion function. The SBS on another hand starts with the complete set of features for consideration, and successively removes the least relevant features according to the criterion function until a specified stop condition is satisfied [205, 230, 231]

Let

$X_k = x_i : 1 \leq i \leq k, x_i \in Y$, be the set of k features

$Y = y_i : 1 \leq i \leq D$, set of D available features

$J(y_i)$, feature selection criterion function of only the ith feature, $y_i, i = 1, 2, \cdots D$

$S_o(y_i)$, value of $J(y_i)$, called individual significance of the feature

The significance $S_{k-1}(x_j)$ of the feature x_j, $j = 1, 2, \cdots, k$ in the set X_k, is

$$S_{k-1}(x_j) = J(X_k) - J(X_k - x_j)$$

The significance $S_{k+1}(f_j)$ of the feature f_j from the set $Y - X_k$ is given by

$$S_{k+1}(f_j) = J(X_k + f_j) - J(X_k)$$

Where,

$$Y - X_k = \{f_i : i = 1, 2, \cdots, D - k, f_i \in Y, f_i \neq x_i \text{ for all } x_i \in X_k\}$$

For $k = 1$, the feature significance equals individual significance. Feature x_j in the set X_k is the most significant (best) feature in the set X_k if

$$\begin{aligned} S_{k-1}(x_j) &= \max_{1 \leq i \leq k} S_{k-1}(x_i) \\ &\implies J(X_k - x_j) \\ &= \min_{1 \leq i \leq k} J(X_k - x_i) \end{aligned}$$

It is the least significant (worst) feature in the set X_k if

$$\begin{aligned} S_{k-1}(x_j) &= \min_{1 \leq i \leq k} S_{k-1}(x_i) \\ &\implies J(X_k - x_j) \\ &= \max_{1 \leq i \leq k} J(X_k - x_i) \end{aligned}$$

Also, feature f_j from the set $Y - K_k$ is the most significant (best) feature with respect to the set X_k if

$$S_{k+1}(f_j) = \max_{1 \leq i \leq D-k} S_{k+1}(f_i)$$

$$\implies J(X_k + f_j)$$

$$= \max_{1 \leq i \leq D-k} J(X_k + f_i)$$

And the least significant feature with respect to the set X_k if

$$S_{k+1}(f_j) = \min_{1 \leq i \leq D-k} S_{k+1}(f_i)$$

$$\implies J(X_k + f_j)$$

$$= \min_{1 \leq i \leq D-k} J(X_k + f_i)$$

Sequential Forward Floating Selection, SFFS

Employs the SFS, sequential forward search [231] and successive removal of worst features, provided an improvement can be made to the feature set [229]. Given a set of features X_k with k features, chosen from the set of features $Y = \{y_j | j = 1, 2, \cdots, D\}$, with a specified criterion function $J(X_k)$.

Step 1. Using SFS, select feature x_{k+1} from the set of available measurements, $Y - X_k$, the most significant feature x_{k+1} with respect to the set X_k is added to X_k to form feature set X_{k+1}

$$X_{k+1} = X_k + x_{k+1}$$

Step 2. Find the least significant features in set X_{k+1}. If x_{k+1} is the least significant feature in the set X_{k+i}, i.e.

$$J(X_{k+1} - x_{k+1}) \geq J(X_{k+1} - x_j), \forall j = 1, 2, ..., k$$

Then set $k = k + 1$ and return to Step 1, but if x_r, $1 \leq r \leq k$ is the least significant feature in the set X_{k+1}, i.e.

$$J(X_{k+1} - x_r) > J(X_k)$$

Then exclude x_r from X_{k+1} to form a new feature set X'_k, i.e.

$$X'_k = X_{k-1} - x_r$$

By now $J(X'_k) > J(X_k)$. If $k = 2$, set $X_k = X'_k$ and $J(X_k) = J(X'_k)$, and return to Step 1, else go to Step 3.

Step 3. Find the least significant feature x_s in the set X'_k.

If $J(X'_k - x_s) \leq J(X_{k-1})$,

then set $X_k = X'_k$, $J(X_k) = J(X'_k)$, and return to Step 1.

If $J(X'_k - x_s) > J(X_{k-1})$,

then exclude x_s from X'_k to form a newly reduced set X'_{k-1}, i.e.

$$X'_{k-1} = X'_k - x_s$$

Set $k = k - 1$.

If $k = 2$,

then set $X_k = X'_k$ and

$J(X_k) = J(X'_k)$, and

return to Step 1,

else repeat Step 3.

Sequential Forward Floating Selection with Multiple Roots, SFFS-MR

Modifying the sequential forward floating selection, Lopes et al 2010, proposed the SFFS-MR, i.e. the sequential forward floating search with multiple roots to better identify genes presenting intrinsically multivariate properties without worsening the asymptotical computational cost of the SFFS. Features (a pair of features) are described as intrinsically multivariate if their predictive properties are synergistic - i.e. together they perform well in predicting a target or class than if considered individually. The SFFR-MR includes multiple roots, typically the best and the worst single results of the SFS algorithm. It was experimentally shown to perform better than the SFS and SFFS methods.

Sequential Forward Floating Selection with Structural Properties (SFFS-BA)

Exploring prior knowledge - topological properties, such as the scale-free property associated with biological networks Lopes et al 2011 further proposed the SFFS-BA method for feature selection [203]. The scale free property describes a disproportionate distribution of node degrees, approximated by a power law distribution - many nodes have a low degree while a few have a high degree [232–238]. From complex network theory, the individual in-degree and out-degree of genes in a network can be used to describe the global network. On the one hand, a uniformly-random Erdos-Renyi (ER) network with randomly connected vertices, assuming that complex systems are connected at random and leads to a Poisson degree distribution with peak near the average degree. On the other hand, a scale-free network is characterized by a power-law in its degree distribution i.e. the probability $P(k)$

of a gene to interact with other k other genes decays as the power law

$$P(k) \sim k^{-\gamma}$$

Where γ is a numeric constant.

Sequential Backward Floating Selection, SBFS

Analogous to the SFFS, but employing the SBS, sequential backward search [230] and in this case successive inclusion of most significant features from available features, provided an improvement can be made to the feature set [229]. Given that k features have already been removed from the complete set of measurements $\bar{X}_o = Y$ to form a feature set $\bar{X}_k$ with the criterion function $J(\bar{X}_k)$;

Step 1. Use the basic SBS method to remove feature x_{k+1} from the current set $\bar{X}_k$ to form a reduced feature set $\bar{X}_{k+l}$, i.e., the least significant feature x_{k+1} is deleted from the set $\bar{X}_k$.

Step 2. Find among the excluded features the most significant feature with respect to the set $\bar{X}_{k+l}$. If x_{k+1} is the most significant feature with respect to $\bar{X}_{k+l}$, i.e.

$$J(\bar{X}_{k+l} + x_{k+l}) \geq J(\bar{X}_{k+l} + x_j), \forall j = 1, 2, \cdots, k$$

then set $k = k + 1$ and return to Step 1. If $x_r, 1 \leq r \leq k$, is the most significant feature with respect to the set $\bar{X}_{k+1}$, i.e.

$$J(\bar{X}_{k+1} + x_r) > J(\bar{X}_k)$$

then include x_r to the set $\bar{X}_{k+1}$ to form a new feature set $\bar{X}'_k$, i.e.

$$\bar{X}'_k = \bar{X}_{k+1} + x_r$$

If $k = 2$, then set

$$\bar{X}_k = \bar{X}'_k$$

$$J(\bar{X}_k) = J(\bar{X}'_k)$$

and return to Step 1. Else go to Step 3.

Step 3. Find from among the excluded features the most significant feature x_s with respect to the set $\bar{X}'_k$. If $J(\bar{X}'_k + x_s) \leq J(\bar{X}_{k-1})$, then set

$$\bar{X}'_k = \bar{X}_k J(\bar{X}'_k) = J(\bar{X}_k)$$

and return to Step 1.

If $J(\bar{X}'_k + x_s) > J(\bar{X}_{k-1})$, then add x_s to the set $\bar{X}'_k$ to form a new enlarged set $\bar{X}'_{k-1}$, i.e.

$$\bar{X}'_{k-1} = \bar{X}'_k + x_s$$

then set $k = k - 1$,

If $k = 2$, then set

$$\bar{X}_k = \bar{X}'_k$$

$$J(\bar{X}_k) = J(\bar{X}'_k)$$

and return to Step 1. Else repeat Step 3

2.5.4 Implementations – GENIE, ARACNe, GGM, etc

Tree-based approaches, GENIE

Described as GEne Network Inference with Ensemble of Trees, Huynh-Thu et (2010), used the tree-based ensemble methods Random Forests or Extra-Trees to determine relevant genes that may predict the expression of the target genes. A potential regulatory link is inferred from the importance of an input gene. Aggregated potential regulatory links are used to rank the interactions from which the whole network is reconstructed [221]. Defining a learning sample to infer network as a sample of N measurements:

$LS = X_1, X_2, \cdots, X_N,$

Where $X_k \in \mathbb{R}^p, k = 1, \cdots, N$ is a vector of expression values of all p genes in the kth experiment:

$X_k = (x_k^1, x_k^2, \cdots, x_k^p)^T$

GENIE aims to assign weights $w_{i,j} \geq 0, (i, j = 1, \cdots, p)$ to potential links from any gene i to gene j. The larger the weight, the greater the significance of the link. Algorithmically, for $j = 1$ to p, GENIE

1. Generates the learning sample of input-output pairs for gene j: $LS^j = (X_k^{-j}, x_k^j), k = 1, \cdots, N,$
2. Uses a tree-based feature selection technique on LS^j to compute weights or confidence levels $w_i, \forall_i \neq j,$
3. Aggregates the p individual gene rankings to get the overall rankings of regulatory links.

4. Solving the nonparametric regression problem using regression trees [220]; uses the squared error loss method to find the function f_j which minimizes the error: $\sum_{k=1}^{N}(x_k^j - f_j(X_k^{-j}))^2$.

Mutual information (MI) based approaches, ARACNe

Margolin et al (2006) described the Algorithm for the Reconstruction of Accurate Cellular Networks(ARACNe) which utilizes the information-theoretic algorithm to infer transcriptional networks from high throughput expression data. ARACNe defines an edge as an irreducible statistical dependency between gene expression profiles that cannot be explained as an artifact of other statistical dependencies in the network [215]. Margolin et al assumed such statistical dependency could explain a biological interaction. Although primarily tailored to identify direct regulatory interactions such as that between a transcription factor and a target gene, other types of interactions may also be identified. ARACNe defined the joint probability distribution (JPD) of the steady state expressions of all genes, $P(g_i), i = 1, \cdots, N$, as

$$P(\{g_i\}) = \frac{1}{Z}\exp\left[-\sum_{i}^{N}\phi_i(g_i) - \sum_{i,j}^{N}\phi_{ij}(g_i,g_j) - \sum_{i,j,k}^{N}\phi_{ijk}(g_i,g_j,g_k) - \cdots\right]$$

$$\equiv e^{-H(\{g_i\})}$$

Where

N is the number of genes,

Z is the normalization factor, also called partition function,

$\phi\cdots$ are potentials, and

$H(\{g_i\})$ is the Hamiltonian that defines the system's statistics.

A set of variables are said to interact if and only if the single potential that depends ex-

clusively on these variables is nonzero. ARACNe aims to identify which of these potentials that are nonzero. It employs maximum entropy approximations [239–241] to $P(g_1, \cdots, g_N)$ consistent with known marginals, to specify potentials. To make estimation less complicated, it truncates the above at the pairwise interactions level,

$$H(\{g_i\}) = \sum_{i}^{N} \phi_i(g_i) + \sum_{i,j}^{N} \phi_{ij}(g_i, g_j)$$

It assumes all genes for which $\phi_{ij} = 0$ are mutually non-interacting, including genes that are statistically independent i.e., $P(g_i, g_j) \approx P(g_i)P(g_j)$, as well as genes that do not interact directly but are statistically dependent due to their interaction via other genes, $P(g_i, g_j) \neq P(g_i)P(g_j)$, but $\phi_{ij} = 0$. Summarily, ARACNe uses a modified concept of Mutual Information, MI, a measure of entropy, to determine the pairwise interaction between features and then applies a data processing inequality (DPI), to eliminate indirect interactions [215, 242].

Probabilistic Graphical model approaches, GGM

Gaussian Graphical Models are undirected probabilistic graphical models that allow the identification of conditional independence relationships among the nodes under the assumption of a multivariate Gaussian distribution of the data [243]. It is based on a stable estimation of the covariance (related to the correlation coefficient) between nodes of this distribution. From this, partial correlations ρ, which are estimates of the strength of direct relationships, are inferred. Given a covariance matrix, C, the element C_{ik} of the covariance matrix is related to the correlation coefficient between nodes X_i and X_k, and $\rho_{i,k}$ describes the correlation between nodes X_i and X_k conditional on all the other nodes in the network. ρ_{ik} is related to the inverse of the covariance matrix C, C^{-1} (with elements C_{ik}^{-1}) [244][243]:

$$\rho_{ik} = -\frac{C_{ik}^{-1}}{\sqrt{C_{ii}^{-1} C_{kk}^{-1}}}$$

Typically, a backward edge exclusion procedure is used to select for a model, with a deviance different stopping rule [245]. This involves 1) creating an initial full graph and the covariance matrix, C, 2) computing the partial correlation coefficient matrix ρ_{ik}, searching for the smallest nonzero absolute ρ_{ik}, replace this with zero, and then computing maximum likelihood estimates for covariance matrix C', 3) using the deviance difference, $dev = N \ln(|C|/|C'|)$ to measure the quality of fit for the selected model. $|C|$ is the determinant of C and $N$is the number of samples. 4) If the probability value of $dev \leq \alpha$ (i.e. significance level $\alpha = 0.05$), the model selection is stopped. Otherwise, the edge (i, k) is deleted from the graph G and the process repeats from step 2. The final selected model is an (a conditional) independence graph, where vertices represent genes and edges are relationships between pairs of genes [245].

Hybrid approaches, SVD-CE

Going beyond just maximally varying features across samples or features with the largest range values, Varshavsky et al (2006) proposed an unsupervised feature selection criterion, based on Singular Value Decomposition SVD entropy [246] a. SVD entropy as the name suggests selects features according to their contribution to the entropy (CE) calculated on a leave-one-out basis. Analogous to search methods described above [229–231] Varshavsky et al proposed four implementations, namely simple ranking according to CE values (SR); forward selection by accumulating features according to which set produces highest entropy (FS1); forward selection by accumulating features through the choice of the best CE out of the remaining ones (FS2); backward elimination (BE) of features with the lowest CE [246].

Chapter 3: Materials and Methods

3.1 Datasets

For this dissertation study, we assumed appropriate datasets to be those derived from genetic experiment assays using the RNAi or the CRISPR-Cas9 technology approach. In addition to the availability of cell viability assay data from these experiments, we expected RNA expression profile data to also be available for concordant assayed samples. We anticipated that available data may either be processed or raw data, though preferably raw. For available raw RNA sequence expression profile data, we expected this to be available in the community de-facto standard – FASTQ formats [247].

3.1.1 Transcriptome, RNA Sequencing Assay Data

With a systematic global internet search, we found a number of deep profiling studies exploring genetic interactions and data repositories containing such studies. Notable amongst these are: GenomeCRISPR [248], GenomeRNAi [249–251], and the Project Achilles [122, 252, 253]. From a cursory look, these appeared appropriate, but the lack of concordant expression profile of the entire genome (gene products), made them inadequate for our regulatory network inference study. Although the Project Achilles attempts to systematically catalogue essential genes across genomically characterized cancer cell lines, genome profiles are not of concordant cell lines i.e. the genome profiles were not necessarily from the same cell populations subjected to genetic knockouts. A more focused query of the National Center for Biotechnology Information (NCBI) Gene Expression Omnibus (GEO Datasets) database [254–256] for RNA sequence expression profile data, with the search phrase "siRNA AND cell line AND cancer AND Homo sapiens AND (Synthetic lethal OR Synthetic lethality)" returned a total of 13 database entries (search date: 2018-12-22, 9:19PM). Of these,

only 2 datasets (with accessions GSE56788 and GSE57871) contained a sizeable number of samples profiled. The two RNASeq expression datasets with available viability assay data were retrieved.

GSE56788

Detailed under the BioProject accession PRJNA244587, this consists of a total of 45 assay samples from 15 biosamples, each ran in 3 independent biological replicates. RNA-seq expression profiles were acquired by next-generation sequencing of vorinostat-resistant HCT116 cells, following knockdown of potential vorinostat-resistance candidate genes. Expression profiles were compared to mock transfection (control). The authors of the study sought to understand the mechanisms by which these knockdowns contributed to vorinostat response. They employed the siRNA-mediated knockdown of each of previously identified resistance candidate genes in the HCT116-VR cell line. For additional details, including phenotype assays description, please see the publication, Falkenberg et al (2014) [39]. Raw RNA sequence expression data were downloaded from the NCBI Sequence Read Archive [257,258], with accession number SRP041162. Table 3.2 shows the transcriptome expression profile sample data accessions and associated siRNA treatment experiments.

GSE57871

The GSE57871 study accession is a 42 sample dataset and an expression profiling by high throughput sequencing. It consists of independent biological experiments of 14 samples performed in triplicates. RNA-seq high throughput expression profiling of vorinostat-resistant HCT116 cells was performed following gene knockdown of GLI1 or PSMD13 with or without vorinostat treatment. Study authors had chosen GLI1 and PSMD13 as potential vorinostat resistance genes because they had previously identified these through a genome-wide synthetic lethal RNA interference screen (the GSE56788 dataset study). An aim was to understand the transcriptional events underpinning the effect of GLI1 and PSMD13 knockdown (sensitisation to vorinostat-induced apoptosis). The authors first performed a knockdown

Table 3.1: GSE56788 Gene Expression Omnibus, GEO dataset I

	GEO_Accession	Experiment	Treatment
1	GSM1369063	SRX516754	mock
2	GSM1369064	SRX516755	siBEGAIN
3	GSM1369065	SRX516756	siCCNK
4	GSM1369066	SRX516757	siCDK10
5	GSM1369067	SRX516758	siDPPA5
6	GSM1369068	SRX516759	siEIF3L
7	GSM1369069	SRX516760	siGLI1
8	GSM1369070	SRX516761	siJAK2
9	GSM1369071	SRX516762	siNFYA
10	GSM1369072	SRX516763	siPOLR2D
11	GSM1369073	SRX516764	siPSMD13
12	GSM1369074	SRX516765	siRGS18
13	GSM1369075	SRX516766	siSAP130
14	GSM1369076	SRX516767	siTGM5
15	GSM1369077	SRX516768	siTOX4
16	GSM1369078	SRX516769	mock
17	GSM1369079	SRX516770	siBEGAIN
18	GSM1369080	SRX516771	siCCNK
19	GSM1369081	SRX516772	siCDK10
20	GSM1369082	SRX516773	siDPPA5
21	GSM1369083	SRX516774	siEIF3L
22	GSM1369084	SRX516775	siGLI1

Table 3.2: GSE56788 Gene Expression Omnibus, GEO dataset II

	GEO_Accession	Experiment	Treatment
23	GSM1369085	SRX516776	siJAK2
24	GSM1369086	SRX516777	siNFYA
25	GSM1369087	SRX516778	siPOLR2D
26	GSM1369088	SRX516779	siPSMD13
27	GSM1369089	SRX516780	siRGS18
28	GSM1369090	SRX516781	siSAP130
29	GSM1369091	SRX516782	siTGM5
30	GSM1369092	SRX516783	siTOX4
31	GSM1369093	SRX516784	mock
32	GSM1369094	SRX516785	siBEGAIN
33	GSM1369095	SRX516786	siCCNK
34	GSM1369096	SRX516787	siCDK10
35	GSM1369097	SRX516788	siDPPA5
36	GSM1369098	SRX516789	siEIF3L
37	GSM1369099	SRX516790	siGLI1
38	GSM1369100	SRX516791	siJAK2
39	GSM1369101	SRX516792	siNFYA
40	GSM1369102	SRX516793	siPOLR2D
41	GSM1369103	SRX516794	siPSMD13
42	GSM1369104	SRX516795	siRGS18
43	GSM1369105	SRX516796	siSAP130
44	GSM1369106	SRX516797	siTGM5
45	GSM1369107	SRX516798	siTOX4

on cells, and then treated these with vorinostat or the solvent control. Two timepoints for drug treatment were assessed: a time-point before induction of apoptosis (4hrs for siGLI1 and 8hrs for siPSMD13) and a timepoint when apoptosis could be detected (8hrs for siGLI1 and 12hrs for siPSMD13). For additional details, including phenotype assays description, see the publication, Falkenberg et al (2016) [40]. Raw sequence expression data were downloaded from the NCBI Sequence Read Archive, accession number SRP042158. Table 3.3 shows the transcriptome expression profile sample data accessions and associated siRNA treatment and treatment timepoint experiments.

3.1.2 Colon Cancer-Associated Genes from OMIM

A curated list of colon cancer-associated genes (Table 3.4) were retrieved from the Online Mendelian Inheritance Man (OMIM) database[259, 260].

3.1.3 Biomedical Significance Experiment Data

To evaluate the clinical and biomedical significance of inferred regulatory features and themes, gene expression profile were retrieved from the cancer genome atlas (TCGA) colorectal cancer mRNA data, in the TCGAcrcmRNA R Bioconductor package[261, 262]. The package contains the TCGA consortium-provided level 3 data, generated by the HiSeq and GenomeAnalyzer platforms, from 450 primary colorectal cancer patient samples[263]. For a more comprehensive and up-to-date phenotype information, associated patients' clinical data were retrieved from the genomic data commons[264–267].

Table 3.3: GSE57871 Gene Expression Omnibus, GEO dataset

	GEO_Accession	Experiment	Timepoint	siRNARx	DrugRx
1	GSM1395357	SRX548949	4hr	mock	DMSO
2	GSM1395358	SRX548950	8hr	mock	DMSO
3	GSM1395359	SRX548951	12hr	mock	DMSO
4	GSM1395360	SRX548952	4hr	mock	vorinostat
5	GSM1395361	SRX548953	8hr	mock	vorinostat
6	GSM1395362	SRX548954	12hr	mock	vorinostat
7	GSM1395363	SRX548955	4hr	siGLI1	DMSO
8	GSM1395364	SRX548956	8hr	siGLI1	DMSO
9	GSM1395365	SRX548957	4hr	siGLI1	vorinostat
10	GSM1395366	SRX548958	8hr	siGLI1	vorinostat
11	GSM1395367	SRX548959	8hr	siPSMD13	DMSO
12	GSM1395368	SRX548960	12hr	siPSDM13	DMSO
13	GSM1395369	SRX548961	8hr	siPSMD13	vorinostat
14	GSM1395370	SRX548962	12hr	siPSMD13	vorinostat
15	GSM1395371	SRX548963	4hr	mock	DMSO
16	GSM1395372	SRX548964	8hr	mock	DMSO
17	GSM1395373	SRX548965	12hr	mock	DMSO
18	GSM1395374	SRX548966	4hr	mock	vorinostat
19	GSM1395375	SRX548967	8hr	mock	vorinostat
20	GSM1395376	SRX548968	12hr	mock	vorinostat
21	GSM1395377	SRX548969	4hr	siGLI1	DMSO
22	GSM1395378	SRX548970	8hr	siGLI1	DMSO
23	GSM1395379	SRX548971	4hr	siGLI1	vorinostat
24	GSM1395380	SRX548972	8hr	siGLI1	vorinostat
25	GSM1395381	SRX548973	8hr	siPSMD13	DMSO
26	GSM1395382	SRX548974	12hr	siPSDM13	DMSO
27	GSM1395383	SRX548975	8hr	siPSMD13	vorinostat
28	GSM1395384	SRX548976	12hr	siPSMD13	vorinostat
29	GSM1395385	SRX548977	4hr	mock	DMSO
30	GSM1395386	SRX548978	8hr	mock	DMSO
31	GSM1395387	SRX548979	12hr	mock	DMSO
32	GSM1395388	SRX548980	4hr	mock	vorinostat
33	GSM1395389	SRX548981	8hr	mock	vorinostat
34	GSM1395390	SRX548982	12hr	mock	vorinostat
35	GSM1395391	SRX548983	4hr	siGLI1	DMSO
36	GSM1395392	SRX548984	8hr	siGLI1	DMSO
37	GSM1395393	SRX548985	4hr	siGLI1	vorinostat
38	GSM1395394	SRX548986	8hr	siGLI1	vorinostat
39	GSM1395395	SRX548987	8hr	siPSMD13	DMSO
40	GSM1395396	SRX548988	12hr	siPSDM13	DMSO
41	GSM1395397	SRX548989	8hr	siPSMD13	vorinostat
42	GSM1395398	SRX548990	12hr	siPSMD13	vorinostat

Table 3.4: Colon-cancer associated genes

	SYMBOL	GENENAME	ENTREZID
1	PLA2G2A	phospholipase A2 group IIA	5320
2	NRAS	NRAS proto-oncogene, GTPase	4893
3	BUB1	BUB1 mitotic checkpoint serine/threonine kinase	699
4	CTNNB1	catenin beta 1	1499
5	PIK3CA	phosphatidylinositol-4,5-bisphosphate 3-kinase catalytic subunit alpha	5290
6	MAPK3	mitogen-activated protein kinase 3	5595
7	MAPK1	mitogen-activated protein kinase 1	5594
8	MAP2K1	mitogen-activated protein kinase kinase 1	5604
9	SMAD2	SMAD family member 2	4087
10	SMAD3	SMAD family member 3	4088
11	SMAD4	SMAD family member 4	4089
12	FGFR3	fibroblast growth factor receptor 3	2261
13	TLR2	toll like receptor 2	7097
14	APC	APC regulator of WNT signaling pathway	324
15	MCC	MCC regulator of WNT signaling pathway	4163
16	PTPN12	protein tyrosine phosphatase non-receptor type 12	5782
17	KRAS	KRAS proto-oncogene, GTPase	3845
18	BRAF	B-Raf proto-oncogene, serine/threonine kinase	673
19	DLC1	DLC1 Rho GTPase activating protein	10395
20	PDGFRL	platelet derived growth factor receptor like	5157
21	RAD54B	RAD54 homolog B	25788
22	PTPRJ	protein tyrosine phosphatase receptor type J	5795
23	CCND1	cyclin D1	595
24	MLH3	mutL homolog 3	27030
25	AKT1	AKT serine/threonine kinase 1	207

3.2 Methods

3.2.1 RNA Sequence Analyses

Quality assessment

For data quality assessment (QA), the fastqcr, ngsReports and Rqc R/bioconductor tools [262, 268–270], modeled after the FASTQC [271] tool philosophy were used. These provide add-on capabilities and the R programming interface to the standalone Java program implementation of FASTQC. QA results were used to identify data with questionable measured quality metrics. In addition to data file statistics, reported quality metrics included;

Table 3.5: Colon-cancer associated genes continued

	SYMBOL	GENENAME	ENTREZID
26	PTEN	phosphatase and tensin homolog	5728
27	BUB1B	BUB1 mitotic checkpoint serine/threonine kinase B	701
28	TP53	tumor protein p53	7157
29	FLCN	folliculin	201163
30	AXIN2	axin 2	8313
31	DCC	DCC netrin 1 receptor	1630
32	BAX	BCL2 associated X, apoptosis regulator	581
33	SRC	SRC proto-oncogene, non-receptor tyrosine kinase	6714
34	AURKA	aurora kinase A	6790
35	EP300	E1A binding protein p300	2033
36	MSH2	mutS homolog 2	4436
37	MLH1	mutL homolog 1	4292
38	PMS1	PMS1 homolog 1, mismatch repair system component	5378
39	PMS2	PMS1 homolog 2, mismatch repair system component	5395
40	MSH6	mutS homolog 6	2956
41	TGFBR2	transforming growth factor beta receptor 2	7048
42	MUTYH	mutY DNA glycosylase	4595
43	CHEK2	checkpoint kinase 2	11200
44	GALNT12	polypeptide N-acetylgalactosaminyltransferase 12	79695
45	SMAD7	SMAD family member 7	4092
46	GREM1	gremlin 1, DAN family BMP antagonist	26585
47	POLD1	DNA polymerase delta 1, catalytic subunit	5424
48	POLE	DNA polymerase epsilon, catalytic subunit	5426
49	WNT1	Wnt family member 1	7471
50	GSK3B	glycogen synthase kinase 3 beta	2932
51	GSK3A	glycogen synthase kinase 3 alpha	2931
52	BCL9	BCL9 transcription coactivator	607

'adapter content', 'overrepresented sequences', 'per base N content', 'per base sequence content', 'per base sequence quality', 'per sequence GC content', 'per sequence quality score', 'sequence duplication levels', and 'sequence length distribution'.

Reads quantification

To quantify expression, we aligned reported reads from the sequencing experiment to the genome. Although non-alignment based quantification approaches such as those implemented in Salmon [272], Sailfish [273], and Kallisto [274] are becoming more popular, the

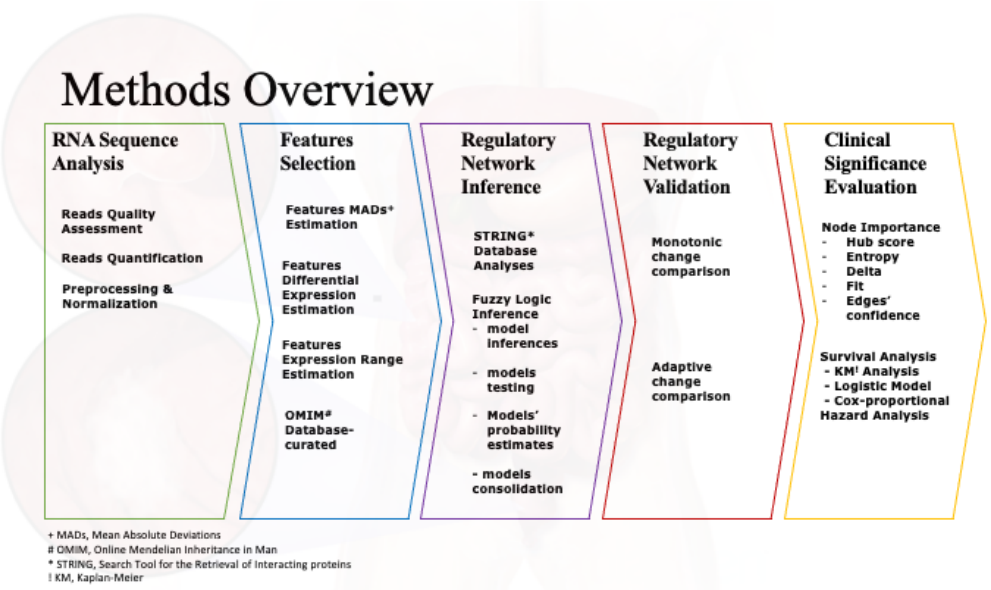


Figure 3.1: Methods Overview

performance of these on quantifying lowly expressed genes and small RNAs is still being debated [275]. Therefore sequence reads were aligned to the genome (NCBI GRCh38 build) using the TopHat2 [276–278] tool which accounts for slice junctions in alignments. Tophat2 uses the bowtie2 [279], noted for its speed and proven memory efficiency for primary alignment. Rather than build new index files, pre-built bowtie2 index files were downloaded from Illumina's iGenomes archive [280]. Accepted hits and annotation information in the BAM format [281] output files were assembled into an expression matrix of feature counts using the featureCount routine in the Rsubread package [282].

Preprocessing and normalization

Feature counts were normalized using the DESeq2 package [283] tool's implemented regularized log transformation to account for disparate total read counts in the different files and to allow for comparison across the different samples. The regularized log transformation moderates the high variance typically observed at low read counts. We specified regularized log transformation intercept as the average expression profile across the normal (mock) samples.

3.2.2 Model Building and Independent Validation Datasets

Datasets were divided into training (regulatory-model-infering) and test (regulatory-model-validation) datasets (Figure 3.2). Regulatory models were inferred using the training datasets. Inferred models were tested in the independent validation datasets. Independent validation dataset included two parts. A part was used to test the regulatory models while the other part was used to test and evaluate a simulation of the consolidated network.

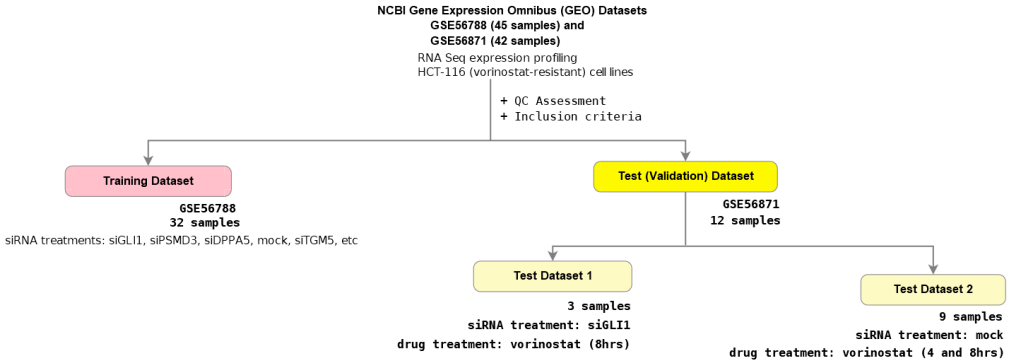


Figure 3.2: Datasets. For our fuzzy-logic inference and evaluation, Two qualifying datasets, with accession numbers GSE56788 and GSE56871, were found and retrieved from the NCBI Gene Expression Omnibus (GEO) database. The studies' samples were subjected to quality assessment and inclusion criteria. 32 qualifying samples from the GSE56788 dataset were used for training (model building) and 12 samples meeting our inclusion criteria from the GSE56871 dataset were used for testing. Of the 12 samples, 3 samples from the 12 were derived from GLI siRNA knockdown experiments and 9 samples were from mock experiments.

3.2.3 Feature Selection

Although similar, feature selection for regulatory network reconstruction and inference differs from classical feature selection. Classical feature selection [193–197] approaches aim to identify the optimal set of features with which a trained model can best predict or correctly identify a class of a not-previously-seen object, given the object's attributes – the class

prediction problem. With a class prediction problem is an associated feature redandancy [202] which needs to be mitigated when choosing an optimal set. With respect to selecting features for regulatory networks however, this may not necessarily be the case, since features that appear redundant may imply co-regulatory (direct or indirect regulatory) interactions in the network. In both situations anyways, on a one hand is the cost of learning a model while on another hand is the curse of dimensionality that plague the low sample to feature ratio characteristic of biological experiments. The very high dimension coupled with low sample size and the potential noise in measured experiments present a limitation for regulatory network inference methods [203] in particular. Feature selection seeks to find a middle spot where cost is minimized with minimal loss in learned model benefits. Although optimized algorithms may mitigate cost, poorly selected or less optimal set of features are set to undermine the efficiency of any learned model.

For a regulatory network model that would represent colon cancer, we reasoned that network features should very likely include known and previously identified products of genes associated with the disease process. Thus, we compiled a list of genes consisting of a curated set obtained from the OMIM database [259,260] and those from literature evidences i.e. genes in described pathways of colon cancer tumorigenesis. And, if we assume that the regulary network is a function of changes in features' expression across time, among different perturbations or across cellular states, it should also appeal to reason that features with significant variations or dispersion in expression across samples should be more informative i.e. more relevant for deriving a regulatory network than those without or with minimal variations. Mathematically, we may describe a cellular state s, as a linear combination of weighted features' expressions, given by the equation below:

$$f(s) = \alpha x_1 + \beta x_2 + \gamma x_3 \cdots + \omega x_n + \epsilon \tag{3.1}$$

where $\alpha, \beta, \gamma, \cdots \omega$ are the **rates of change** in respective feature's expression i.e. *rate constants*; ϵ is the random error estimate; $\{x_1, x_2, x_3 \cdots x_n\}$ is the set of expression values of

features under condideration; and n is the total number of features. We reasoned that if we assume a regulatory network describes changes in cellular state across time, we might as well describe it as a first derivative of cellular state, $f(s)'$. Therefore features without changes in expression across time, i.e. features whose rate constants tended to zero would drop off in the estimate $d(f(s))/dt$. This is analogous to being of less significance in determining the dynamic nature of the regulatory network, i.e. changes in cellular state.

To determine maximally varying features, from our RNA sequence analyses normalized expression values, we estimated a mean absolute deviation (MAD) from the mean, for each feature. Given by,

$$\frac{1}{n}\sum_{i=1}^{n}|x_i - \bar{x}| \tag{3.2}$$

where n in this case is the number of samples or perturbations and $\bar{x}$ is the mean expression value of the specific feature across the samples. $x_i \in \{x_1, x_2, ...x_n\}$.

To further assess variation in the expression of genes across samples, we also determined fold changes between the minimum and maximum expression values for for the respective genes and the strength of change between knockdown and control experiments. Because genes with highest MADs were observed to be predominantly those with low average expression and thus may be confounded by a Poisson noise distribution, we performed differential expression analyses between the respective groups of knockdown (siRNA) experiments and the controls to identify statistical significantly expressed genes (i.e. features with true changes)[284–286].

In summary, in additon to genes previously identified as related to colon cancer tumorigenesis and the specific genes targeted in the knockdown experiments, expression profile-informed genes were also considered for regulatory network inference based on their MAD, differential expresson and the log fold difference between the minimum and maximum expression values across siRNA knockdown experiments. The expresion profile-based selection

criteria we specified were that for a gene to be considered:

1. Its mean absolute deviations (MAD) must be greater than the median of MADs.

2. Its expression value in 80% of samples must be greater than its minimum value across all samples by a minimum of two folds. The 80% of samples must include $\geq$ 80% of siRNA-targeted experiments. And, it must be

3. Statistical siginificant and differentially expressed in at least two siRNA-targeted sample groups versus the control group

Knowledge-guided feature selection

Purely data-driven methods have drawbacks such as limited biological interpretability. Likewise, canonical signaling pathways from literature evidences, provided in curated knowledge databases are not very specific and these hardly predict cell type-specific responses to experimental situations [287]. Therefore, we employed a hybrid approach that addresses these limitations and, can integrate prior knowledge and real data for network inference. We searched the derived features, and the colon cancer related gene features from OMIM database, against the STRING database[288–290]. Our search parameters included: a search against a full network type where edges indicate both functional and physical protein interactions; reported network edges indicate the presence of evidence of interactions between nodes; active interaction sources included mining of literature texts (TextMining), known experiments, knowledge bases, documented co-expression information, gene neighborhood, fusion and co-occurrence information. Quantitative interaction score for retrieved edges was specified as a minimum of 0.150. We retrieved features reported to be part of a potential network. For each feature found as part of a potential network, all reported interacting features were retrieved and mapped. We elaborated regulatory relationships between and among features using the fuzzy logic approach.

3.2.4 Fuzzy Logic Regulatory Models Inference

To tease regulatory interactions among our initial selection of features, we employed the fuzzy-logic approach. The fuzzy logic approach mitigates known challenges of modeling biological systems, such as inconsistencies and inaccuracies associated with high-throughput characterizations. These challenges also include data noise and those of dealing with a semi-quantitative data [175]. Similar to Boolean networks, fuzzy logic methods are simple and are fit to model imprecise and or highly complex networks. And, opposed to differential equation based models, they are less computationally expensive and less sensitive to imprecise measurements [176–178]. Fuzzy logic compensates for the inadequate dynamic resolution of a Boolean (or discrete) network, while simultaneously addressing the computational complexity of a continuous network [179, 180].

A significant advantage of the fuzzy logic approach is that, in contrast to many other automated decision making algorithms or regulatory inference methods, such as neural networks or polynomial fits, algorithms in fuzzy logic are presented in similar day-to-day conversational language. Therefore, a fuzzy logic is more easily understood and can be extrapolated in predictable ways.

In general, the fuzzy logic modeling approach entails three major steps (Fig. 3.3):

1. Fuzzification
2. Rule evaluation, and
3. Defuzzification

[181].

Fuzzification

Considering expression as a linguistic variable and applying defined membership functions on observed continuous numerical expression data, the fuzzification step derives qualitative

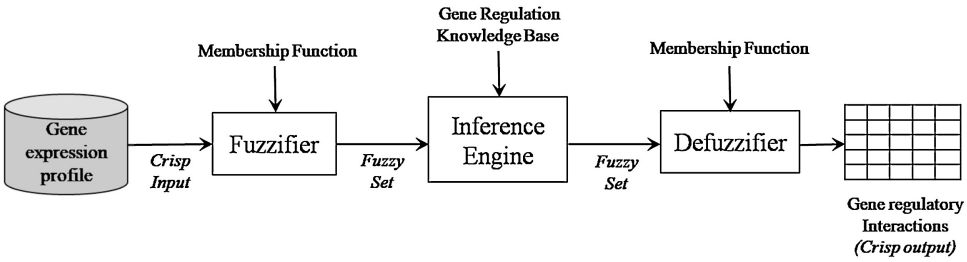


Figure 3.3: A generic pipeline of fuzzy logic model of GRN inference(Raza 2019)

values. It is a mapping of non-fuzzy inputs to fuzzy linguistic terms [181]. To make data fuzzification easier, a normalization technique may be applied to scale values to within a preferred range [179, 181, 182].

The fuzzification step derives qualitative values from the expression profile's crisp values. By applying defined membership functions on crisp, numerical expression data, we derived qualitative values – described as a mapping of non-fuzzy inputs to fuzzy linguistic terms [291]. Given qualitative values of HIGH, MEDIUM, or LOW, the fuzzification step takes a feature's expression value and assigns it degrees to which it belongs to the respective class of HIGH, MEDIUM or LOW expression values. [292–295]. After an initial data transformation of log2 expression ratios by the arctan function and dividing values by $\frac{\pi}{2}$, to project the ratios onto $[-1, 1]$, the fuzzification step utilizes three membership functions consisting of the 'low', 'medium', and 'high' functions. Given the three fuzzification functions ($y_1 = \text{low}$, $y_2 = \text{medium}$, $y_3 = \text{high}$), fuzzification of a gene expression value x results in the generation of a fuzzy set $y = [y_1, y_2, y_3]$ as follows:

$$y_1 = \begin{cases} x, x < 0 \\ 0, x \geq 0 \end{cases} \tag{3.3}$$

$$y_2 = 1 - |x|, \forall x \tag{3.4}$$

$$y_3 = \begin{cases} 0, x \leq 0 \\ x, x > 0 \end{cases} \tag{3.5}$$

$$\tag{3.6}$$

Rule evaluation

The rule evaluation step considers combinations of features and utilizes an inference engine of rules, of the form IF-THEN, including fuzzy set operations such as AND, OR, or NOT, to evaluate input features' expression (in fuzzy set definition) in relation to output features. This has been described as attempting to make an expert judgment of collective linguistic terms; attempts to find a solution to an evaluation of the concurrent state of existence of linguistic description of states.

We specified our rule configuration (the specification of if-then relationships between variables in fuzzy space) in the form of a vector $r = [r_1, r_2, r_3]$. We specified the state of an output node $z = [z_1, z_2, z_3]$ to be determined by the fuzzy state of an input feature $y = [y_1, y_2, y_3]$ and the rule describing the relationship between the input and the output, $r = [r_1, r_2, r_3]$ as follows:

$$z = [y_{r1}, y_{r2}, y_{r3}] \tag{3.7}$$

An inhibitory relationship, for example, specified as $[3, 2, 1]$ implies, if input is low (r_1),

then output is high (3); if input is medium (r_2), then output is medium (2), and if input is high (r_3), then output in low (1). The classic fuzzy logic rule evaluation using the logical AND connective results in a combinatorial rule explosion i.e. an exponential increase in the number of rules to be evaluated and computational time, with additional inputs to be considered [83]. Therefore, to address this combinatorial rule explosion situation, we employed the logical OR (union) rule configuration, an algebraic sum in fuzzy logic [154,296] as described in [49].

Defuzzification

The defuzzification step produces a quantifiable expression result or value given the input sets, the fuzzy rules, and membership functions. Defuzzification technically interpretes the membership degrees of the fuzzy sets into a specific decision or real value. The defuzzification step attempts to report a corresponding continuous numerical variable from a fuzzy state liguistic variable. Several approaches to defuzzify abound. We employed the simplified centroid method [296]. Given a predicted fuzzy values of an output node $y = [y_1, y_2, y_3]$, we defined defuzzified expression values ($\bar{x}$) as:

$$\bar{x} = \frac{y_3 - y_1}{y_1 + y_2 + y_3} \tag{3.8}$$

After defuzzification, we reverse transformed back to log2 expression values by multiplying derived values by $\frac{\pi}{2}$ and applying the tangent function.

Inferred regulatory model fit

For each regulatory model, which consists of an output feature, its suggested regulatory input feature(s) and associated fuzzy logic rules (relating each input feature to the output respectively), we estimated the fitness of such model's prediction of the output x across M experiment samples or perturbations $x = \{x_1, x_2, ..., x_M\}$ as:

$$E = 1 - \frac{\sum_{i=1}^{M}(x_i - \tilde{x}_i)^2}{\sum_{i=1}^{M}(x_i - \bar{x})^2} \tag{3.9}$$

$$\tag{3.10}$$

where $\tilde{x} = \{\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_M\}$ is the set of defuzzified numerical log expression ratios predicted for the output feature and $\bar{x}$ is the mean of the experimental values of x across the samples or perturbations observed. A perfect fit would result in a maximum E of 1.0.

Model probability (p-value) estimates

To estimate models probabilities, we fitted a probability density distribution for $100,000$ fit estimates of models derived by random permutations of rules and input features for each output features. We allowed up to four regulatory interactors. We computed a model fit's p-value as the probability of observing an estimated fit from a random estimated fits distribution. A gamma distribution was fitted and, the 'scale' and 'shape' parameters were derived using The Maximum Likelihood Estimate (MLE) approach [297–300] implemented in the egamma function, in the EnvStat R package. With the "scale' and 'shape' parameters, random deviates and cummulative probabilities were derived using the (rgamma) and (pgamma) implementations respectively, in the stats package [301, 302].

Model validation

As described above, the fuzzy logic approach infers a regulatory model to consist of an output node, input nodes and respectively derived regulatory rule that relate each input node to the output node. We validated derived models for each feature output in the independent *GLI1* siRNA knockdown experiments datasets generated by Falkenberg et al (2016). In this dataset, the authors focused on the genes *GLI1* and *PSMD13* as potential vorinostat-resistance candidate genes, identified from previous screens. Falkenberg and colleagues performed transcriptome analysis on vorinostat-resistant HCT116 cells

Table 3.6: Independent Validation Dataset

	siRNARx	drugRx	timepoint
SRX548958	siGLI1	vorinostat	8hr
SRX548972	siGLI1	vorinostat	8hr
SRX548986	siGLI1	vorinostat	8hr

(HCT116-VR) upon knockdown of these candidate genes in the presence and absence of vorinostat. According to the authors, treatment of vorinostat-resistant cells with the *GLI1* small-molecule inhibitor, GANT61, **phenocopied** the effect of *GLI1* knockdown. Therefore, for independent validation of our inferred regulatory models, we reason that for model estimated fit in the test data should as closely as possible be similar to (or better than the) estimated fit in the training dataset. The two timepoints for drug treatment assessed by Falkenberg and colleague represent a timepoint before induction of apoptosis (4hrs for siGLI1) and a timepoint when apoptosis could be detected (8hrs for siGLI1). Therefore for this validation, we used the sample expression data at 8hrs (see the table 3.6).

3.2.5 Network Construction and Validation

For each output node, the best-fitted model as determined by estimated fit difference between the associated models in the training and validation data was selected as a representative model. Representative models were consolidated into a single regulatory network (Figure 3.4). We reasoned that, models with minimal estimated fit difference are more likely stable than those with high differences.

Network validation

To validate the derived regulatory network, we compared the monotonic and adaptive changes[303] observed by a dynamic simulation of the network over $5,000$ time-step iterations in the training data against that observed in the validation data. We reasoned that the distribution of observed changes between the training data network simulation and the independent validation data simulation would not be significantly different.

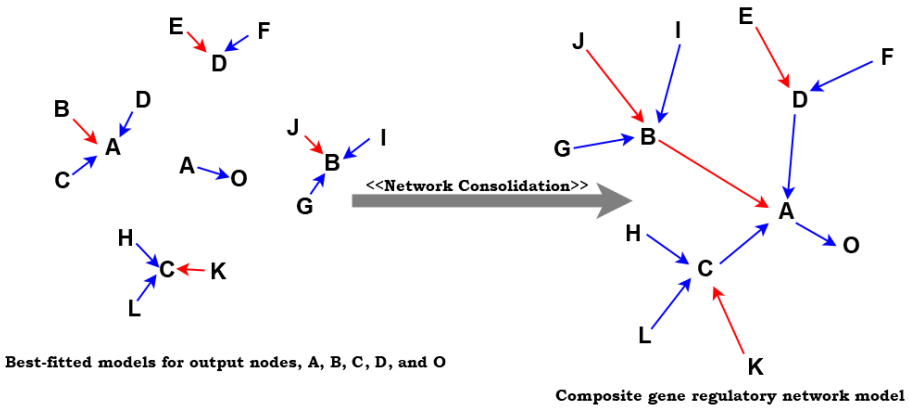


Figure 3.4: Regulatory network construction – constructed from consolidation of representative best-fitted models for all output nodes

To simulate the network, we derived successive time-step expression values (I_{n+1}) for each node by a linear combination of the previous (I_{n-1}) and new values (I_n), to ensure the system converge smoothly towards equilibrium[294]. Given by Gormley et al, new values (I_n) were computed as:

$$I_{n+1} = \alpha I_n + (1 - \alpha) I_{n-1} \tag{3.11}$$

Where the α option specifies the 'mixing parameter', guiding how quuickly the simulation reaches system equilibrium. New values for each node were based on the initial conditions and the fuzzy relations (regulatory rules) inferred from the training data. Zhang et al (2019) respectively described monotonic S_M and adaptive changes S_A as:

$$S_M = \frac{|R_T - R_0|}{\max(R)} \tag{3.12}$$

$$S_A = \frac{\max(|R - R_0|}{|R_T - R_0|} \tag{3.13}$$

Where R are the estimated values over the entire iteration, R_0 are observed values at the start of simulation and R_T are values observed at the end of simulation. We utilized the Student t-test to determine if there is any difference in monotonic and adaptive network simulation changes between the training data and independent network validation data. To effectively simulate a knockdown and making the validation dataset-2 more comparable, we in-silico kept the level of knocked-down feature expression unchanged throughout the simulation steps. The table (Table 3.7) shows the dataset considered for independent validation of regulatory network (validation dataset-2).

Table 3.7: Independent Validation Dataset for in-silico knockout network simulation

	siRNARx	drugRx	timepoint
SRX548952	mock	vorinostat	4hr
SRX548953	mock	vorinostat	8hr
SRX548954	mock	vorinostat	12hr
SRX548966	mock	vorinostat	4hr
SRX548967	mock	vorinostat	8hr
SRX548968	mock	vorinostat	12hr
SRX548980	mock	vorinostat	4hr
SRX548981	mock	vorinostat	8hr
SRX548982	mock	vorinostat	12hr

3.2.6 Biomedical Significance Evaluation

Node importance

To evaluate biomedical significance of inferred regulatory network, we first estimated importance of all nodes contained therein. We defined node importance score (I_i) similar

to Zhang et al's [303]. The node importance score estimates integrate network topology, network edge interaction strengths and gene expression. To encapsulate these, Zhang and colleagues defined a hub score (H), a local network entropy (S) and an adaptation score (A) and integrated these into a comprehensive index for each node – a normalized rank sum of these values.

A Hub score assesses a node's connectivity to other nodes. It is the principal eigenvector of the adjacency matrix of the inferred regulatory network. If

$$H = (h_1, h_2, \cdots h_L) \tag{3.14}$$

Zhang et al described the hub score of node i as h_i.

Extending the works of Teschendorff and Severini [304], Zhang et al described local entropies as the degree of randomness in the local pattern of information flux around each node[303]. This is analogous to the centrality entropy described by Ortiz-Arroyo and Hussein [305]. It is a measure of the centrality of nodes depending on their contribution to the entropy of the derived regulatory network. We computed each nodes local entropy using Jalili et al's centiserve R package implementation of entropy[306]; derived from Shannon's [307] definition of entropy which states that the entropy of a random variable X that can take n values is:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \tag{3.15}$$

Jalili et al's centrality entropy measure H_{ce} of a graph G, is defined as:

$$H_{ce}(G) = -\sum_{i=1}^{n} \gamma(v_i) \times \log_2 \gamma(v_i) \tag{3.16}$$

where $\gamma(v_i) = \frac{paths(v_i)}{paths(v_1,v_2,...,v_M)}$ where $paths(v_i)$ is the number of geodesic paths from node v_i to all the other nodes in the graph and $paths(v_1, v_2, ..., v_M)$ is the total number of geodesic paths M that exists across all the nodes in the graph.

In place of an adaptation score rank, we modified the node importance score to include instead the fit rank (r^F), the mean edges confidences rank (r^E) and the delta rank (r^D). We defined the fit rank as the rank of the estimated fit associated with the respective node in the network. We defined the mean edges confidences rank as the rank of the average of edge confidences returned from the STRING database associated with the node and contained in the node's regulatory model inferred by the fuzzy logic approach. To moderate the estimated fits, we defined the delta rank as the rank of the difference in model-associated estimated fits observed in the training and independent validation datasets.

We defined an importance score (I_i) for each node as the normalized rank sum of these values, similar to Zhang et al's.

$$I_i = \frac{r_i^H + r_i^S + r_i^F + r_i^E + r_i^D}{\sum_{i=1}^{L}(r_i^H + r_i^S + r_i^F + r_i^E + r_i^D)} \tag{3.17}$$

Logistic regression and survival analysis

Similar to Zhang and colleagues'[303], we evaluated the potential for highly ranked regulatory node features or themes to predict short- (three or less years) and mid-term survival (greater than 3 years). We reasoned that these features are potentially able to drive tumor cells to either circumvent or succumb to epistatic events. We fitted a logistic regression model using the expression profile and clinical information we retrieved on the cancer genome atlas (TCGA) primary colorectal cancer samples – incorporating our derived node importance measures as penalty weights and specifying the 3-year survival statuses (dead or alive) as the outcome. Given $y_i = 0$ or 1 as the binary response outcome associated with the i-th sample in n patients; $p_i = \Pr(y_i = 1)$; $i = 1, \cdots, n$; and $x_i = (x_{i1}, x_{i2}, \cdots, x_{iL})^T$ is the expression profiles of the genes in the i-th patient, we modeled the logistic regression

model as:

$$\text{logit}(p_i) = \log\left[\frac{\Pr(y_i = 1)}{1 - \Pr(y_i = 1)}\right] = \beta_0 + \sum_{j=1}^{L} \beta_j x_{ij} \tag{3.18}$$

where β_0 and β_j are respectively the intercept and regression coefficients.

We randomly divided the data into training and test subdatasets at varying sample ratios of 50%, 60%, 70%, and 80%. We ran 100 repeated estimates at the different sample ratios. We calculated the areas under the ROC curves (AUCs) for the training and test dataset. We further evaluted the association of the top ranked features with survival using a Kaplan–Meier (K-M) survival analysis[308–310] and estimated significance between the K-M curves using the Cox proportional hazard model[311] and the two-sided log-rank test[312]. We classified patients into two groups (high-risk vs low-risk) based on the optimal cutoff using the ROC approach.

3.2.7 Tools and implementations

The fuzzy logic regulatory inference method of Gormley et al [182] and its optimization for quicker time to inference is implemented in the platform independent Java programming language. Its source codes and pre-compiled binaries are freely available at the repository locations:

- https://github.com/paiyetan/jfuzzymachine

- https://github.com/paiyetan/jfuzzymachine/releases/tag/v1.7.21

- https://bitbucket.org/paiyetan/jfuzzymachine/src/master/

- https://bitbucket.org/paiyetan/jfuzzymachine/downloads/

All statistical analyses were done in the R programming and computational statistics environment [269]. Network diagrams and analyses were also done using the Cytoscape tool, version 3.8.2 running on a Mac OS X 10.15.7 – x86_64 operating system.

Chapter 4: Results

4.1 RNA Sequence Analyses

From a total of 45 samples in the GSE56788 dataset, 34 samples succesfully passed through our analysis pipeline. 11 samples failed because of potentially corrupted raw data files. Samples that passed are shown in Table 4.1. These include two assays each of the interferring RNA treatment samples siCCNK, siEIF3L, siGLI1, siJAK2, siNFYA, siPOLR2D, siPSMD13 and siRGS18; one assay of the siBEGAIN treatment sample; and, three assays each for the interferring RNA treatment samples siCDK10, siDPPA5, siSAP130, siTGM5 and siTOX4. Two assays were experiment control samples. Spannig gene products involved in the cell cycle-, gene transcription- and signal transduction pathways-associated biological processes, Table 4.2 shows the siRNA targeted (knocked-down) genes in the vorinotat-resistant colon cancer cell line sythetic lethality experiment assays.

4.1.1 Reads quality assessment

All quality assessment measures as defined by Andrews and colleagues at the Babraham Institute[271], except the 'Per base sequence content' module which represents the relative amount of each base in the entire genome (Table 4.3), were satisfied. These included the: 'Adapter Content', 'Overrepresented sequences', 'Per base N content', 'Per base sequence content', 'Per base sequence quality', 'Per sequence GC content', 'Per sequence quality scores', 'Sequence Duplication Levels, and the 'Sequence Length Distribution' module passed QC assessment. In most instances, the 'Per base sequence content' is not of biological concern as this arises from technical issues relating to using primers with random hexamers or the use of transposases which are biased toward specific cleavage sites during the library generation step. It is understood that, because of this some biases may occur,

Table 4.1: GSE56788 dataset QC assessment samples

	Sample	Treatment
1	SRX516756.sra_data	siCCNK
2	SRX516757.sra_data	siCDK10
3	SRX516758.sra_data	siDPPA5
4	SRX516759.sra_data	siEIF3L
5	SRX516760.sra_data	siGLI1
6	SRX516761.sra_data	siJAK2
7	SRX516762.sra_data	siNFYA
8	SRX516763.sra_data	siPOLR2D
9	SRX516764.sra_data	siPSMD13
10	SRX516765.sra_data	siRGS18
11	SRX516766.sra_data	siSAP130
12	SRX516767.sra_data	siTGM5
13	SRX516768.sra_data	siTOX4
14	SRX516769.sra_data	mock
15	SRX516772.sra_data	siCDK10
16	SRX516773.sra_data	siDPPA5
17	SRX516781.sra_data	siSAP130
18	SRX516782.sra_data	siTGM5
19	SRX516783.sra_data	siTOX4
20	SRX516784.sra_data	mock
21	SRX516785.sra_data	siBEGAIN
22	SRX516786.sra_data	siCCNK
23	SRX516787.sra_data	siCDK10
24	SRX516788.sra_data	siDPPA5
25	SRX516789.sra_data	siEIF3L
26	SRX516790.sra_data	siGLI1
27	SRX516791.sra_data	siJAK2
28	SRX516792.sra_data	siNFYA
29	SRX516793.sra_data	siPOLR2D
30	SRX516794.sra_data	siPSMD13
31	SRX516795.sra_data	siRGS18
32	SRX516796.sra_data	siSAP130
33	SRX516797.sra_data	siTGM5
34	SRX516798.sra_data	siTOX4

particularly at the start of the sequence reads[271]. The 'Per base sequence content' failure is triggered if the difference between A and T, or G and C is greater than 20% in any position. This is generally not a problem if the biases and failures can be visualized (see Figure 4.1) and attributed to around the first 12 base locations.

Table 4.2: siRNA Experiments Targeted Genes

	SYMBOL	GENENAME	ENTREZID
1	BEGAIN	brain enriched guanylate kinase associated	57596
2	CCNK	cyclin K	8812
3	CDK10	cyclin dependent kinase 10	8558
4	DPPA5	developmental pluripotency associated 5	340168
5	EIF3L	eukaryotic translation initiation factor 3 subunit L	51386
6	GLI1	GLI family zinc finger 1	2735
7	JAK2	Janus kinase 2	3717
8	NFYA	nuclear transcription factor Y subunit alpha	4800
9	POLR2D	RNA polymerase II subunit D	5433
10	PSMD13	proteasome 26S subunit, non-ATPase 13	5719
11	RGS18	regulator of G protein signaling 18	64407
12	SAP130	Sin3A associated protein 130	79595
13	TGM5	transglutaminase 5	9333
14	TOX4	TOX high mobility group box family member 4	9878

Figure 4.2 shows the Per Base Sequence Quality plot for the SRX516756.sra_data. This shows the distribution of quality scores for bases at the respective positions in a box plot with whiskers. The y-axis shows the quality scores. A better base call is indicated by a higher score. The background of the graph divides the y axis into very good quality (green), reasonable quality (orange), and poor quality (red) calls. It is not unusual for the quality of a base call to degrade toward the end of the read. Although scores appear generally okay, it can be seen that the scores reported at the first $5-10$ base positions are of lesser quality.

Table 4.4 shows estimates for quality assessment parameters in each RNA sequencing sample in the GSE56788 dataset. Assessed parameters included percent duplication, percent GC content, and average sequence read length. Average duplication rate is estimated to be 20.21%, while that of GC content stood at 48.68%.

4.1.2 Reads quantification

Maximum number of reads was found to be 26430192 reads in the SRX516756.sra_data (siCCNK) sample, while the minimum was 9756110, in the SRX516790.sra_data (siGLI1) sample - an approximately three fold difference across the sythetic lethal experiment assays

Table 4.3: GSE56788 dataset QC assessment by quality control measures

	measure	samples	fail	pass
1	Adapter Content	34	0	34
2	Basic Statistics	34	0	34
3	Overrepresented sequences	34	0	34
4	Per base N content	34	0	34
5	Per base sequence content	34	34	0
6	Per base sequence quality	34	0	34
7	Per sequence GC content	34	0	34
8	Per sequence quality scores	34	0	34
9	Sequence Duplication Levels	34	0	34
10	Sequence Length Distribution	34	0	34

(Table 4.5). As opposed to 45 samples, results are presented for 34 samples. As previously mentioned, data for 11 samples failed on topHat2 alignment on execution, potentially due to corrupted samples' raw data file.

Similar QC assessment profile is observed for the data in both the GSE56788 and the GSE57871 datasets.

4.2 Feature Selection, for Regulatory Network Inference

4.2.1 Features' mean absolute deviations (MADs)

As previously described, we reasoned that the most changing features, in terms of expression values across pertubations, are more informative in the context of regulatory networks than non-changing features (see section 3.2.3). To determine these features, we estimated the mean abosolute deviations of each of the $28,395$ genes across all available assay samples in our model-inferring (training) dataset. The observed median of MADs is 0.2033 (Mean=0.2905, SD=0.3482). With a maximum and minimum observed MAD of 2.1333 and 0.0 respectively, over 30% $(11,123)$ of features do not appear to change (Fig 4.3, 1st Qu=0.0000). These include those for knocked-down features, DPPA5, RGS18, and TGM5.

Table 4.4: GSE56788 dataset QC assessment of sequence reads

	Sample	Duplicates(%)	GC%	Length
1	SRX516756.sra_data	23.64	48.00	51
2	SRX516757.sra_data	17.62	48.00	51
3	SRX516758.sra_data	18.78	49.00	51
4	SRX516759.sra_data	19.32	48.00	51
5	SRX516760.sra_data	22.75	48.00	51
6	SRX516761.sra_data	21.38	49.00	51
7	SRX516762.sra_data	17.91	49.00	51
8	SRX516763.sra_data	23.47	49.00	51
9	SRX516764.sra_data	18.53	48.00	51
10	SRX516765.sra_data	20.36	49.00	51
11	SRX516766.sra_data	19.52	49.00	51
12	SRX516767.sra_data	18.08	49.00	51
13	SRX516768.sra_data	20.73	48.00	51
14	SRX516769.sra_data	18.67	49.00	51
15	SRX516772.sra_data	18.30	48.00	51
16	SRX516773.sra_data	18.26	49.00	51
17	SRX516781.sra_data	20.91	49.00	51
18	SRX516782.sra_data	20.16	49.00	51
19	SRX516783.sra_data	19.73	49.00	51
20	SRX516784.sra_data	17.66	49.00	51
21	SRX516785.sra_data	24.11	49.00	51
22	SRX516786.sra_data	23.17	48.00	51
23	SRX516787.sra_data	20.90	49.00	51
24	SRX516788.sra_data	19.87	49.00	51
25	SRX516789.sra_data	19.26	48.00	51
26	SRX516790.sra_data	19.57	49.00	51
27	SRX516791.sra_data	20.42	49.00	51
28	SRX516792.sra_data	18.01	49.00	51
29	SRX516793.sra_data	24.32	49.00	51
30	SRX516794.sra_data	19.15	49.00	51
31	SRX516795.sra_data	18.77	48.00	51
32	SRX516796.sra_data	23.23	49.00	51
33	SRX516797.sra_data	17.94	49.00	51
34	SRX516798.sra_data	22.56	48.00	51

The feature with maximum MAD was HRNR (hornerin).

Table 4.5: GSE56788 dataset sequence reads

	Sample	Treatment	Total
1	SRX516756.sra_data	siCCNK	26430192
2	SRX516757.sra_data	siCDK10	11926761
3	SRX516758.sra_data	siDPPA5	14012687
4	SRX516759.sra_data	siEIF3L	17264395
5	SRX516760.sra_data	siGLI1	16841563
6	SRX516761.sra_data	siJAK2	19008717
7	SRX516762.sra_data	siNFYA	12986269
8	SRX516763.sra_data	siPOLR2D	16712374
9	SRX516764.sra_data	siPSMD13	13782737
10	SRX516765.sra_data	siRGS18	16399848
11	SRX516766.sra_data	siSAP130	14471644
12	SRX516767.sra_data	siTGM5	14561182
13	SRX516768.sra_data	siTOX4	17051622
14	SRX516769.sra_data	mock	14581166
15	SRX516772.sra_data	siCDK10	12726019
16	SRX516773.sra_data	siDPPA5	12870208
17	SRX516781.sra_data	siSAP130	14235470
18	SRX516782.sra_data	siTGM5	13362627
19	SRX516783.sra_data	siTOX4	13970706
20	SRX516784.sra_data	mock	11168125
21	SRX516785.sra_data	siBEGAIN	23372015
22	SRX516786.sra_data	siCCNK	17367956
23	SRX516787.sra_data	siCDK10	15621563
24	SRX516788.sra_data	siDPPA5	14589712
25	SRX516789.sra_data	siEIF3L	14106493
26	SRX516790.sra_data	siGLI1	9756110
27	SRX516791.sra_data	siJAK2	15299261
28	SRX516792.sra_data	siNFYA	11219308
29	SRX516793.sra_data	siPOLR2D	15782493
30	SRX516794.sra_data	siPSMD13	12804888
31	SRX516795.sra_data	siRGS18	12792707
32	SRX516796.sra_data	siSAP130	18050180
33	SRX516797.sra_data	siTGM5	12923634
34	SRX516798.sra_data	siTOX4	17513707

4.2.2 Features' differential expression

Another measure of change we employed was the differential expression for each feature – an estimate of features that are truly different in terms of expression values between conditions. We estimated the differential expression of features in each knock-down assay group

against the control assays. At adjusted p-values ≤ 0.05, the maximum number of differentially expressed features $(8,055)$ were found in the POLR2D siRNA knockdown assays, while the least number $(2,504)$ were found in the RGS18 knockdown experiments (Table 4.6, Fig. 4.4 and 4.5). $1,645$ features were differentially expressed in 5 comparisons of siRNA knockdown assays versus control assays, while 6 features are differentially expressed in all comparisons(Fig. 4.6). The features found to be differentially expressed in all contrasts include the NRBP1, MTHFD2, ALDH1A3, TNS2, LIMA1 and the BAK1 gene products. Cumulatively $13,090$ features were found to be differentially expressed in at least one contrast comparison, while $4,270$ were found to be differentially expressed in at least half of the comparisons (Fig. 4.7, Table 4.7). In terms of features discovered to be differentially expressed, the assay groups appear to cluster into 3 major groups (Fig. 4.8).

Table 4.6: Number of differentially expressed features between siRNA knockdown assays and control assays

	Differentially Expressed Features	At ≤ 0.05 Adjusted P-value
siCCNK	12237	7068
siCDK10	10404	3644
siDPPA5	11838	5476
siEIF3L	10108	2553
siGLI1	12409	6842
siJAK2	11192	4035
siNFYA	10393	3236
siPOLR2D	12032	8055
siPSMD13	11848	5883
siRGS18	10456	2504
siSAP130	12312	7540
siTGM5	11478	5513
siTOX4	11121	3566
siBEGAIN	10878	2528

4.2.3 Features' expression ranges and log-fold changes

Still on evaluating features' expression changes across knockdown experiments, we considered the log-fold change between the minimum and the maximum expression value for each

Table 4.7: Table of cumulative occurrence of features differentially expressed. Cumulatively, $13,090$ features are differentially expressed in at least one comparison while 6 features are differentially expressed in all 14 comparisons between the different knockdown assays versus the control experiment assays

Comparisons	Features
1	13090
2	11753
3	10521
4	9077
5	7519
6	5873
7	4270
8	2871
9	1754
10	922
11	428
12	156
13	47
14	6

feature across all knockdown assays. A large proportion of features show no log-fold change (Fig. 4.9). The maximum log fold change (10.436) is found in the SPP1 feature expression profile. Across all features, median log-fold change was 1.099 while mean log-fold change was 1.497. $8,037$ features have log-fold changes ≥ 2 while only 19 features have log-fold change ≥ 8 (Fig. 4.10 and Table 4.8).

Table 4.8: Number of features with min-max log-fold change with greater than or equal the specified values.

min-max Log-fold	number of features
2	8037
3	5034
4	3213
5	1501
6	448
7	91
8	19

4.2.4 Online Mendelian inheritance in man (OMIM) database features

For a more encompassing regulatory network inference feature set, and addressing limitations of purely data-driven approaches as previously mentioned, 52 features were retrieved from the online mendelian inheritance in man (OMIM) database (Table 3.4)[259, 260].

4.2.5 Search tool for the retrieval of interacting proteins (STRING) database search

According to criteria specified previously (see Methods in Chapter 3), and together with features determined from the OMIM database, 571 were considered to be temporally changing and potentially informative for regulatory network construction. These were searched against the STRING database for any remote biological evidence of potential interactions – serving as a priori knowledge guide for our downstream fuzzy logic regulatory network inference. At a false discovery rate of 0.05 and minimum interation confidence of 0.150 (low confidence), retrieved interaction network consisted of 559 nodes and $8,819$ edges. Average node degree was 31.6 and the average local clustering coefficient was 0.312. With an expected number of edges of $7,659$, p-value of protein-protein interaction (PPI) enrichment was $< 1e{-}16$. With 238 mapped interactions, AKT was reported to have the most identitfied interactions. Features with only one SRING database-identified interaction were C1orf35, CCDC171, MROH8, OR51B2, PRB3, and RNF223.

4.2.6 Selected features for fuzzy logic based regulatory network

Of the 559 features found to belong to probable biological network in the STRING database, 535 were subjected to the fuzzy logic regulatory inference approach.

4.3 Regulatory Network Inference

An inferred fuzzy logic model consists of an output node, its regulatory input nodes, and respective regulary rule that describes the interaction and relationship of the input to the

output node. A regulatory rule is one of 27 rules, each represented by a three-member array notation (or tuple). The indices of the array represent respectively a low, medium and high presupposed state of the input node. Actual value (1, 2, or 3) of each element in the array represents the respective expected state (low, medium, or high) of the output node. For example, the rule [3, 2, 1] states that: when the input expression value is 'low', the output node is 'high'(3); when the input is 'medium', the output is 'medium'(2); and when the input is 'high', the output is 'low'(1). This represents a classic repression-like regulation (i.e. negative control). The converse is true for a rule [1, 2, 3] representation. It implies that when the input expression value is 'low', the output node is 'low'(1); when the input is medium, the output is medium(2); and when the input is 'high', the output is 'high'(1). This represents a classic activation-like regulation (i.e. positive control).

4.3.1 Fuzzy logic-based regulatory models

Filtering at estimated fit of 0.70, 299 output nodes and fuzzy logic regulatory models were obtained. These consist of 402 gene features. Ranked by models' minimum difference between estimated fit in training data and independent validation data, the top models include the output nodes: TGFBR2 (model fit = 0.7011; adjusted p-value = 5.720849e−04), RIMBP3B (model fit = 0.7972; adjusted p-value = 1.907588e−05), PPP2R1A (model fit = 0.7162; adjusted p-value = 9.497439e−05), UBQLN2 (model fit = 0.7770; adjusted p-value = 4.702329e−05), WNT3A (model fit = 0.7300; adjusted p-value = 1.872238e−04), TP53 (model fit = 0.7144; adjusted p-value = 2.021138e−03), and PIK3CA (model fit = 0.7040; adjusted p-value = 7.431795e−04) amongst many others (Tables 4.9, 4.10, and 4.11). Enumerated to regulate the TGFBR2 gene were three regulatory inputs. These include, a inhibitory interaction (fuzzy logic rule, [2, 1, 1]) by the TNS1 gene, a stimulatory interaction (fuzzy logic rule, [1, 1, 3]) by the AXIN2 gene, and another inhibitory interaction (fuzzy logic rule, [3, 2, 2]) by the CCND1 gene products respectively. For the TP53 gene, two stimulatory interactions by the TP53I3 and the WNT1 gene products were identified

(fuzzy logic rules, $[1, 2, 3]$, and $[1, 1, 3]$ respectively) (Tables 4.9, 4.10, and 4.11).

Table 4.9: Top 25 fuzzy-logic regulatory models identified (see text for rule explanation)

	Output	Inputs	Rules
1	TGFBR2	[TNS1, AXIN2, CCND1]	[[2, 1, 1], [1, 1, 3], [3, 2, 2]]
2	RIMBP3B	[GNAZ, RIMBP3]	[[1, 3, 3], [1, 2, 3]]
3	PPP2R1A	[SMAD2, RASGRP3, BCL9]	[[3, 1, 1], [3, 2, 2], [2, 3, 3]]
4	UBQLN2	[ANK1, COL4A5, GAS7]	[[2, 1, 3], [1, 2, 3], [3, 1, 1]]
5	RIMBP3C	[SCN2A, RIMBP3, RIMBP3B]	[[1, 1, 3], [1, 2, 3], [1, 3, 3]]
6	SYNGR3	[KIF3C, RBPMS2, SPTBN4]	[[1, 2, 3], [1, 2, 3], [1, 3, 3]]
7	POLE	[POLE4, MSH2]	[[1, 3, 2], [2, 3, 1]]
8	MAOB	[NRG2, CYP1B1, FAXC]	[[1, 2, 3], [1, 3, 2], [2, 3, 3]]
9	PARVG	[HSPG2, LCP1, NOD2]	[[1, 3, 3], [3, 1, 2], [2, 1, 1]]
10	WNT3A	[AURKA, NTN1]	[[2, 1, 1], [1, 3, 2]]
11	GALNT12	[ST3GAL3, R3HDM2, FLCN]	[[1, 1, 3], [1, 3, 3], [3, 2, 1]]
12	SCN2A	[ANK1, ATRNL1]	[[2, 3, 3], [1, 2, 2]]
13	DDX60	[UBC, PMS1, CMPK2]	[[3, 2, 1], [1, 3, 3], [1, 3, 3]]
14	SERPINA5	[AKT1, CTNNB1, SERPINA1]	[[3, 3, 1], [1, 3, 3], [1, 2, 3]]
15	RIMBP3	[SCN2A, RIMBP3C, RIMBP3B]	[[1, 1, 3], [1, 3, 3], [1, 2, 3]]
16	GPR176	[MC1R, CYP4F22, GNAZ]	[[1, 2, 3], [1, 2, 3], [1, 1, 3]]
17	FLCN	[GALNT12, NRAS, CTNNB1]	[[3, 2, 1], [3, 3, 1], [3, 2, 1]]
18	MYO1D	[MYO7A, LCP1, MGAT5B]	[[2, 1, 1], [1, 1, 2], [2, 3, 3]]
19	KCNQ1	[TP53, PARVG, PIK3R5]	[[1, 3, 3], [1, 2, 3], [1, 1, 3]]
20	CPLX2	[MGAT5B, BEGAIN]	[[1, 2, 3], [1, 2, 3]]
21	TP53	[TP53I3, WNT1]	[[1, 2, 3], [1, 1, 3]]
22	PIK3CA	[PMS1, POLD1, AXIN2]	[[1, 3, 3], [3, 2, 1], [3, 1, 1]]
23	ATRNL1	[KLK5, SCN2A]	[[1, 3, 3], [1, 2, 3]]
24	RIN2	[GPRIN2, KCNK3, MYOM1]	[[1, 2, 2], [1, 3, 3], [2, 2, 3]]
25	LMO7	[TNFRSF19, LIMCH1]	[[1, 2, 3], [1, 2, 3]]

4.3.2 Models consolidation – fuzzy logic-based regulatory network

Combining the derived fuzzy logic models into a consolidated network as previously described, we obtained a network with 402 nodes, 849 edges, and a network mean clustering coefficient of 0.018. Consisting predominantly of out-degrees, the maximum degree of 20 is observed at the TP53 gene feature node, followed closely by the SRC (degrees = 19), LONRF2 (degree = 12), PIK3CA (degree = 11), AKT1 (degree = 11), NTN1 (degree =

Table 4.10: Top 25 fuzzy-logic regulatory models estimated fits

	Model output	Training fit	Test fit
1	TGFBR2	0.70	0.70
2	RIMBP3B	0.80	0.80
3	PPP2R1A	0.72	0.71
4	UBQLN2	0.78	0.77
5	RIMBP3C	0.76	0.75
6	SYNGR3	0.76	0.77
7	POLE	0.72	0.74
8	MAOB	0.72	0.74
9	PARVG	0.72	0.71
10	WNT3A	0.73	0.71
11	GALNT12	0.72	0.74
12	SCN2A	0.71	0.74
13	DDX60	0.71	0.68
14	SERPINA5	0.72	0.75
15	RIMBP3	0.76	0.80
16	GPR176	0.76	0.71
17	FLCN	0.71	0.65
18	MYO1D	0.75	0.69
19	KCNQ1	0.70	0.65
20	CPLX2	0.74	0.68
21	TP53	0.71	0.79
22	PIK3CA	0.70	0.61
23	ATRNL1	0.74	0.64
24	RIN2	0.70	0.81
25	LMO7	0.76	0.63

11), MAPK3 (degree = 10), AURKA (degree = 10), CCND1 (degree = 10), UNC5A (degree = 10), CHEK2 (degree = 10), ICAM1 (degree = 10) and the UBC (degree = 10) gene feature nodes. Average number of neighbors is 3.826, network diameter is 22, characteristic path length is 6.816, and network density is 0.005. Fig. 4.11

4.4 Regulatory Network Validation

Validating regulatory network in independent dataset by topological changes observed on dynamic simulations of inferred regulatory network, no statistical difference is seen in the distribution of monotonic (t statistic $= -1.5104$, p-value $= 0.1315$) and the adaptive changes

Table 4.11: Top 25 fuzzy-logic regulatory models p-values

Model output	adj. p-value (BH)
TGFBR2	5.720849e-04
RIMBP3B	1.907588e-05
PPP2R1A	9.497439e-05
UBQLN2	4.702329e-05
RIMBP3C	2.522838e-05
SYNGR3	1.702524e-04
POLE	9.527495e-03
MAOB	1.862614e-04
PARVG	1.393662e-05
WNT3A	1.872238e-04
GALNT12	2.539112e-03
SCN2A	2.451987e-04
DDX60	1.250976e-04
SERPINA5	7.866536e-05
RIMBP3	1.961288e-05
GPR176	1.117347e-03
FLCN	5.681593e-03
MYO1D	2.346973e-05
KCNQ1	2.345625e-04
CPLX2	1.763330e-05
TP53	2.021138e-03
PIK3CA	7.431795e-04
ATRNL1	3.359898e-05
RIN2	1.815780e-04
LMO7	2.380319e-04

(t statistic = 1.2079, p-value = 0.2278) across all features over a $5,000$ time steps (Fig. 4.12 and 4.13).

4.5 Biomedical Significance Evaluation

4.5.1 Node importance estimation

Based on the characterized fuzzy logic regulatory network, we computed the hubscore, entropy, mean edge confidence, the associated model fit and the delta change (in fit estimate between training and independent test data), for each gene in the regulatory network (see

Methods section) to measure the node importance and to identify key genes driving changes to sensitivity or resistance to Vorinostat in colorectal cancer. Emerging tops with respect to the hubscore, entropy, mean edge confidence, model fit and the delta change scores are the genes TP53, MAGEE1, POLE, TGFBR2 and BUB1 respectively (Supplementary Table). We calculated the normalized score accordingly, to estimate the importance score for each gene (see Methods). Top ranked genes include UBC, PTEN, SMAD2, LMO7, GNAZ, POLR2D, TP53, AKT1, RIMBP3, and CCNK (Table 4.12).

4.5.2 Logistic regression and survival analysis

We reasoned that features driving resistance to vorinostat are very likely drivers of aggressivenes and therefore of poor patient clinical outcomes. We evaluated the potential clinical significance of these vorinostat-resistance associated features in three different ways using colon cancer transcriptomic and clinical data from the cancer genome atlas (TCGA) assayed samples. From the genome data bank retrieved data 334 patient samples had associated clinical information. Of these, 77 samples have had a survival event. The Median time to event is 334.0 days (Mean = 540.7 days), while maximum time to event is 2821.0 days. At different sampling ratios, the 77 samples were randomly divided into a training and a test subdataset and repeating the sample division at each ratio 100 times. Assigning the node importance score of the network feature expression values, Figure 4.16 shows the AUC estimates' distribution in the training data and test data weighted logistic regressions. AUC estimates ranged up to 0.99 and 0.93 in the training and test dataset, suggesting a potential optimal subset. Based on data from the TCGA samples, the top 10 percent (by node importance) of features shows a significant association with the survival probability (log-rank p-value = < 0.0001) of colon cancer patients (Fig. 4.17) – demonstrating the significant clinical relevance of the top identified genes and their roles in cancer progression.

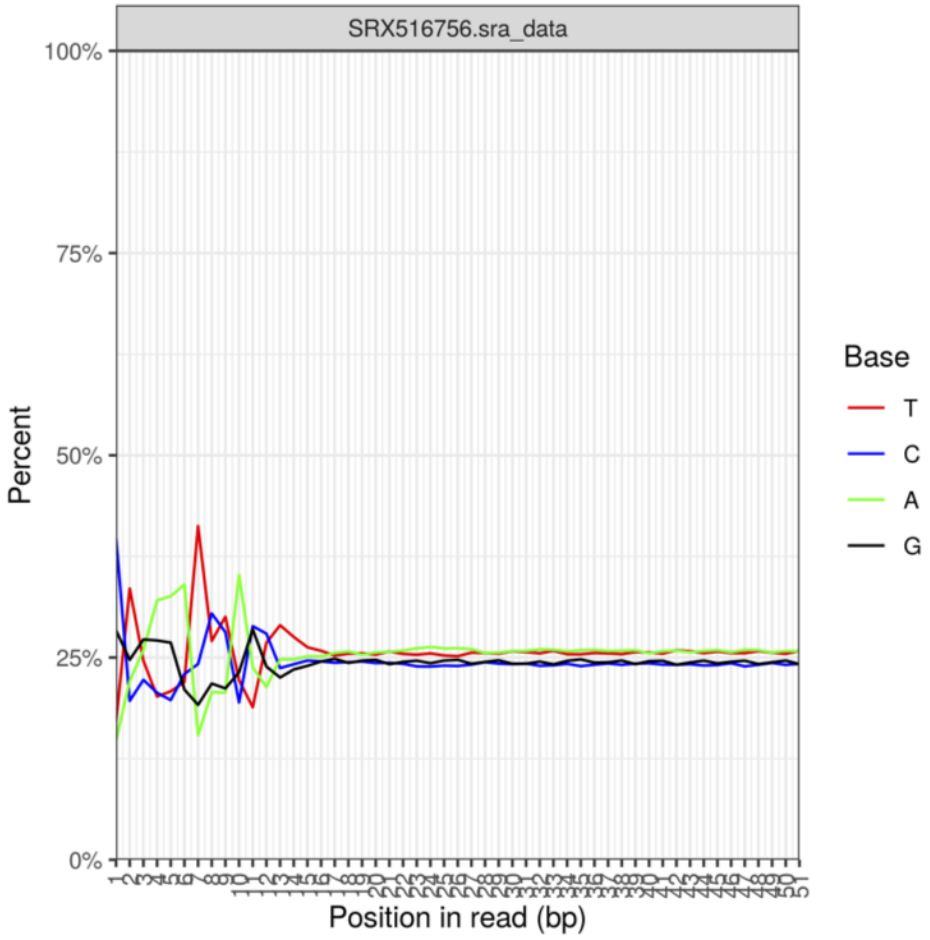


Figure 4.1: Per Base Sequence Content Plot - the proportion of each each of the four normal DNA bases at each sequence read position in the SRX516756.sra_data (siCCNK) experiment sample

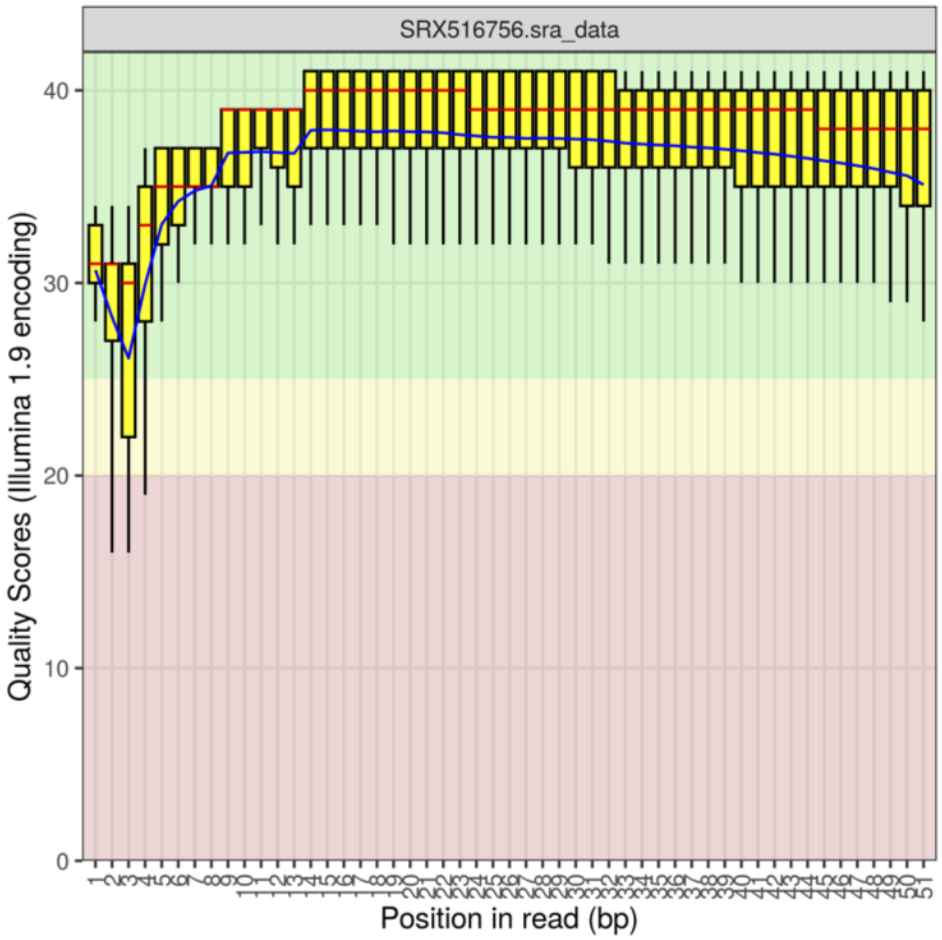


Figure 4.2: Per Base Sequence Quality Plot - the distribution of quality scores for bases at the respective positions in the SRX516756.sra_data experiment sample

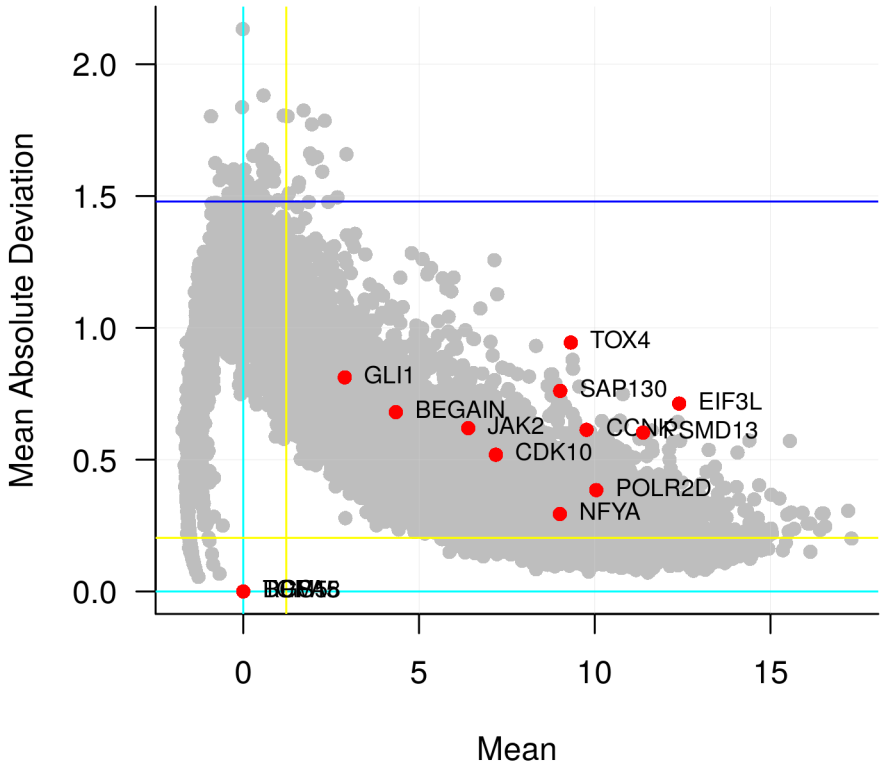


Figure 4.3: A plot of mean absolute deviations versus mean of log expresson values across vorinostat-resistant colon cancer synthetic lethal experiment assays. Coordinates of siRNA-targeted (knocked-down) features in the different assays are indicated in red. The yellow lines indicate the median of the estimated mean absolute deviations and the median of the mean log-expressions respectively.

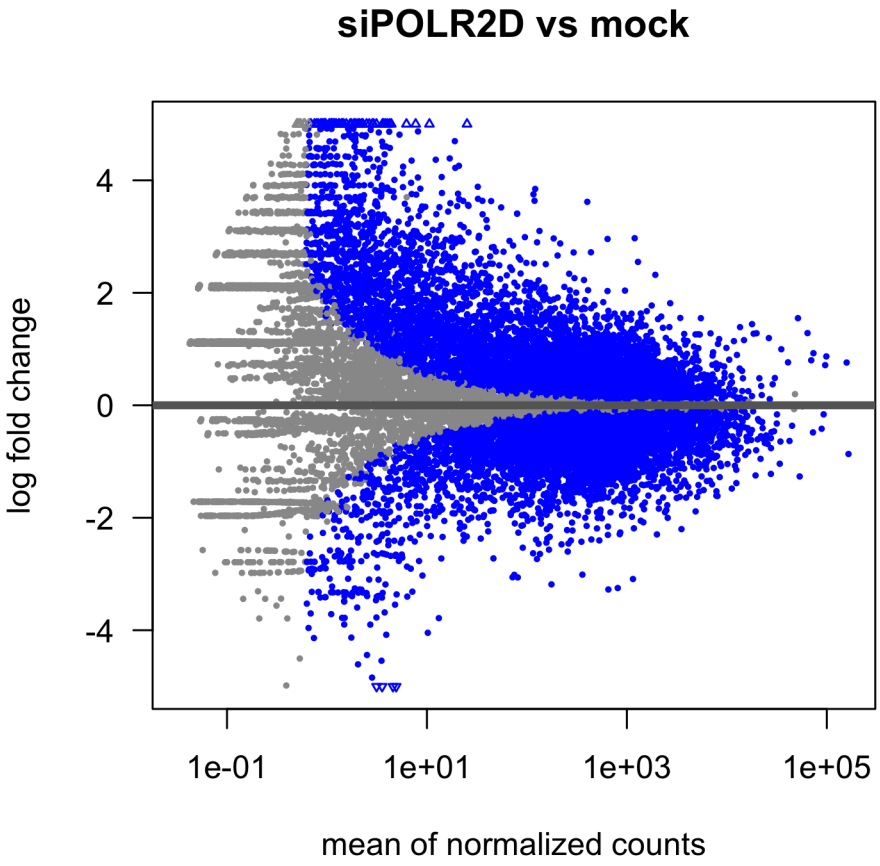


Figure 4.4: MA Plot highlighting computed differentially expressed features (blue) between the POLR2D siRNA knockdown assays and the control (mock siRNA) assays

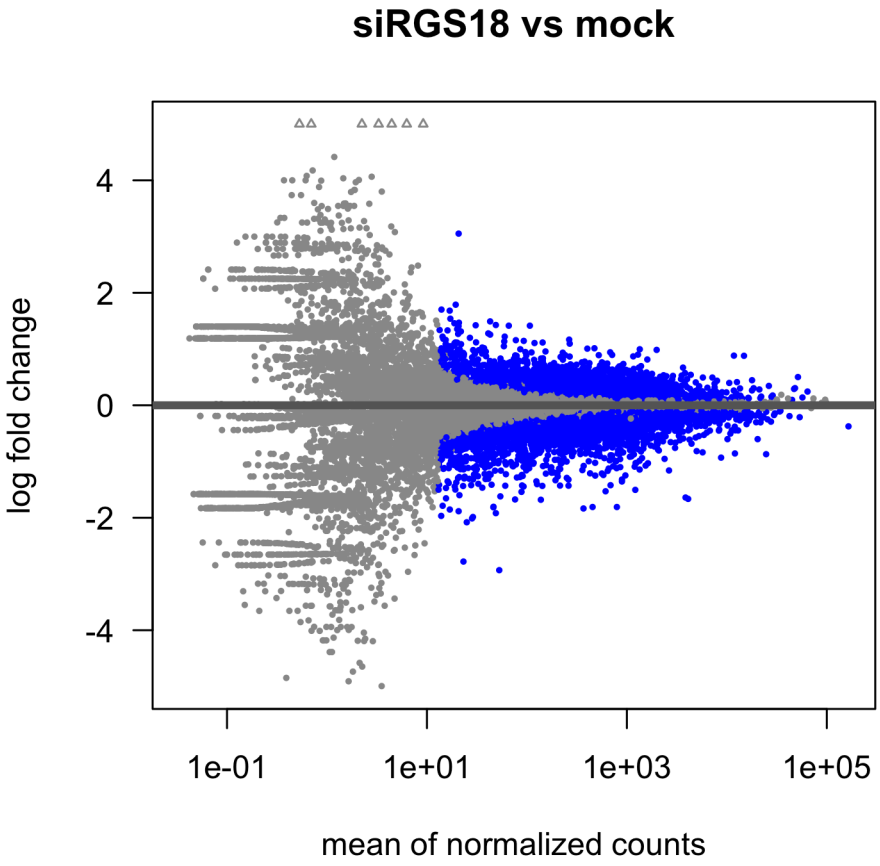


Figure 4.5: MA Plot highlighting computed differentially expressed features (blue) between the RGS18 siRNA knockdown assays and the control (mock siRNA) assays

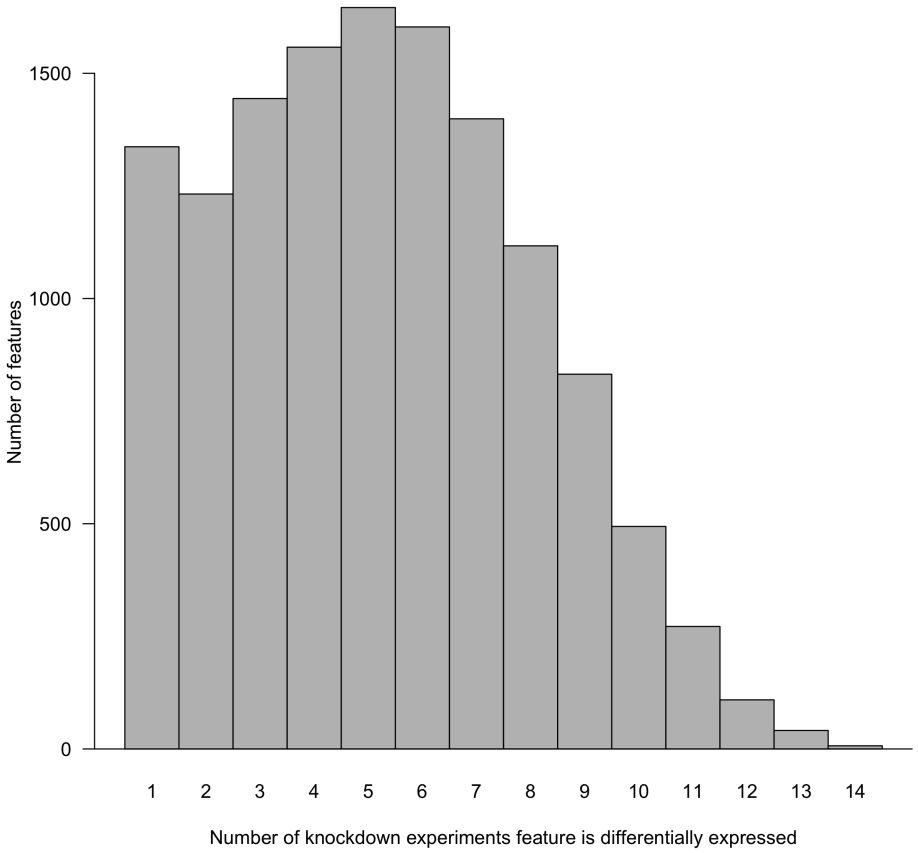


Figure 4.6: A barplot showing the number of features found to be differentially expressed in the comparisons against the control experiments. 6 features are found to be differentially expressed in all 14 comparisons of siRNA knockdown experiments versus the controls. Most of the differentially expressed features are found in a least 5 comparisons.

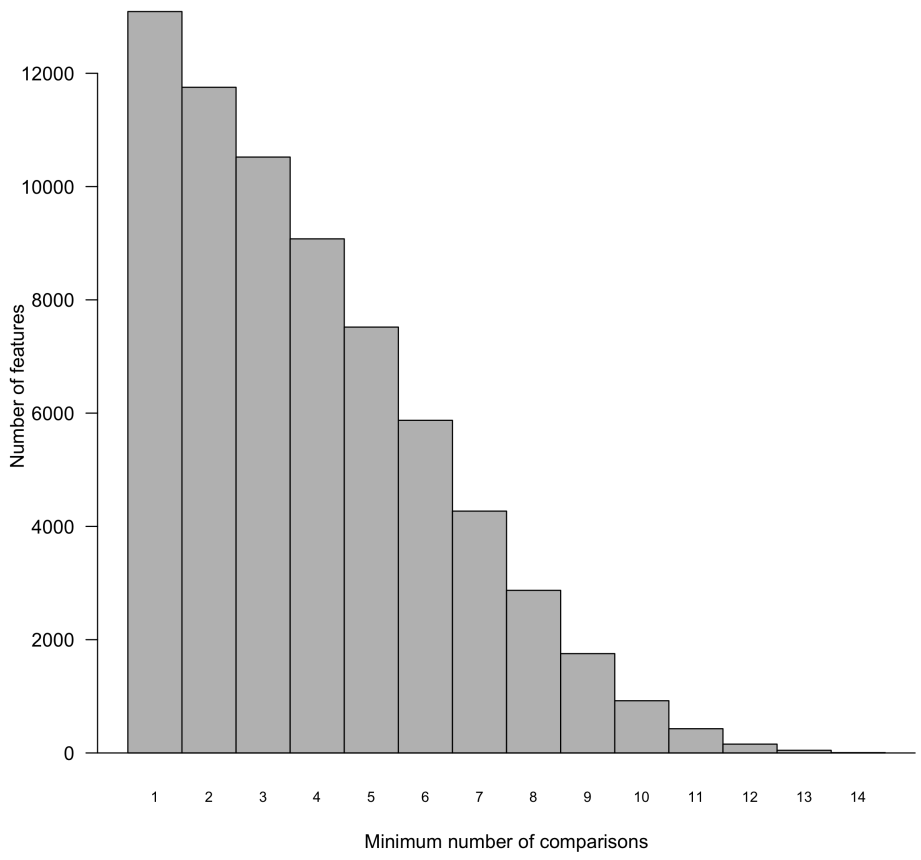


Figure 4.7: Barplot showing the cumulative number of features found to be differentially expressed in the comparisons against the control experiments. 6 features are found to be differentially expressed in all 14 comparisons of siRNA knockdown experiments versus the controls. Cumulatively $13,090$ features were found to be differentially expressed in at least one comparison, while $4,270$ were found to be differentially expressed in at least half of the comparisons

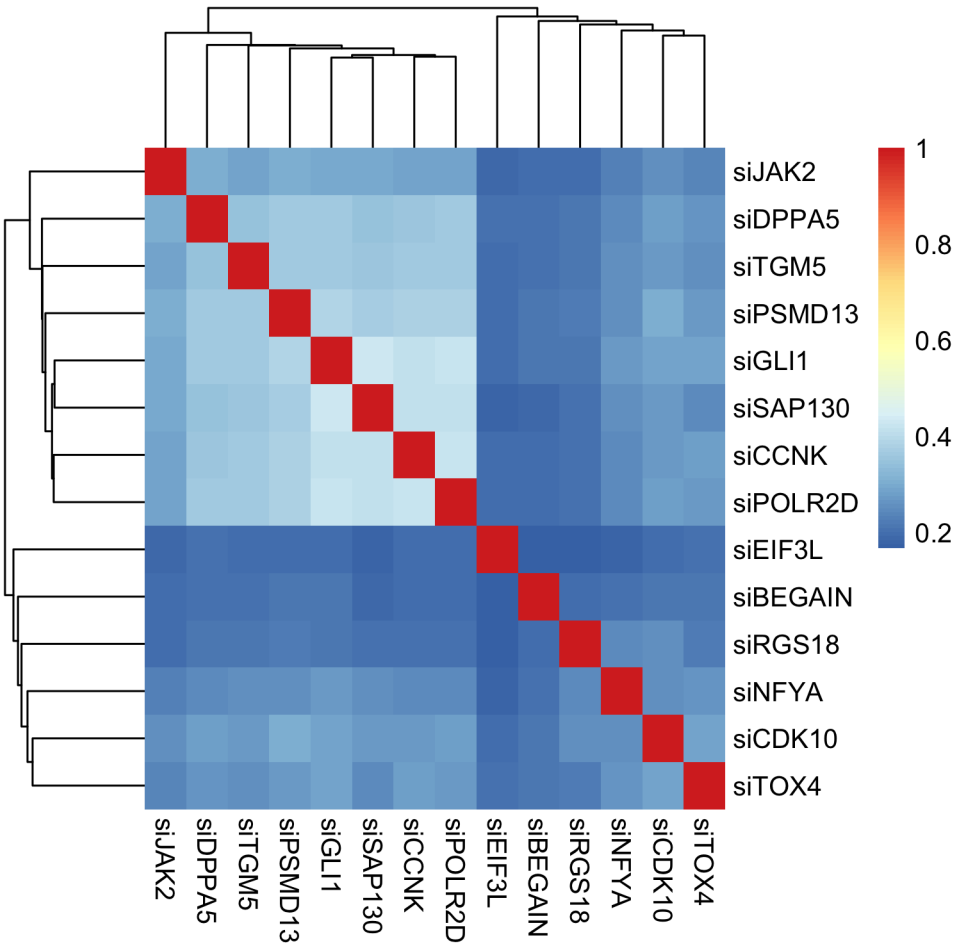


Figure 4.8: A heatmap of similarities between the siRNA knockdown assay groups, in terms of differentially expressed features. The Jaccard score estimate was used as a measure of similarities. The higher the estimated value, the more similar the groups are

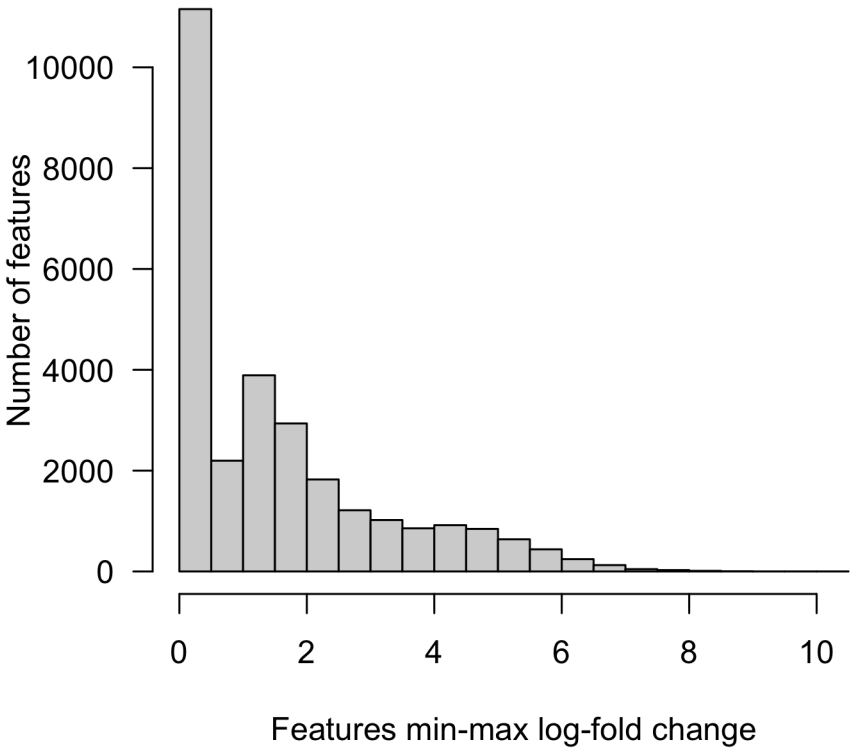


Figure 4.9: A histogram of log-fold changes between the minimum and maximum expression values for features across knockdown experiments.

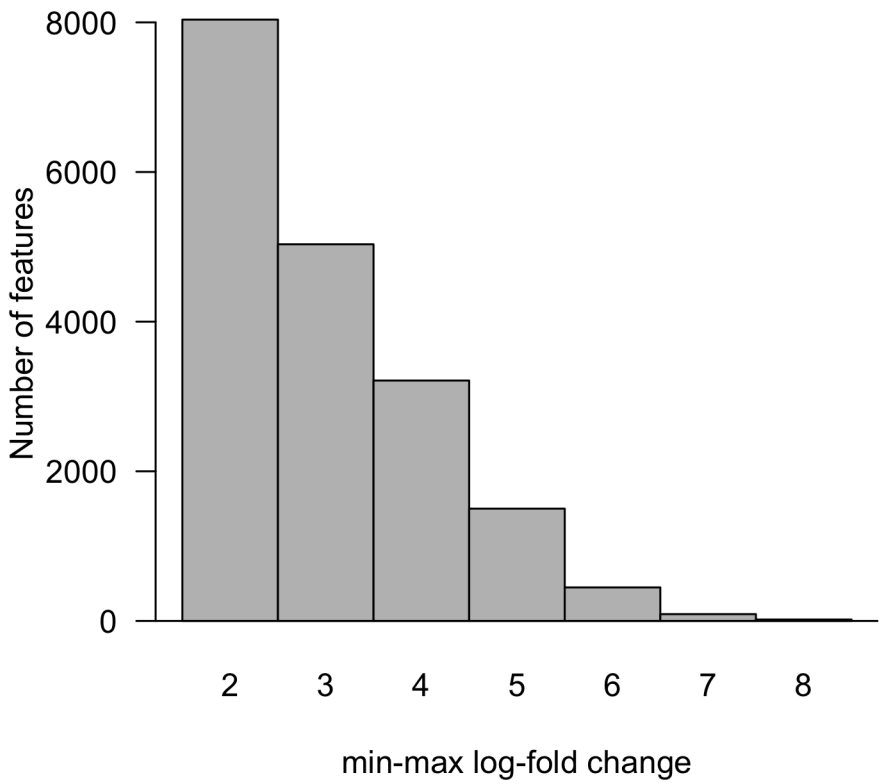


Figure 4.10: A boxplot showing the number of features with greater than or equal a value of the specified log-fold change (difference) between the minimum and maximum expression values across experiments.

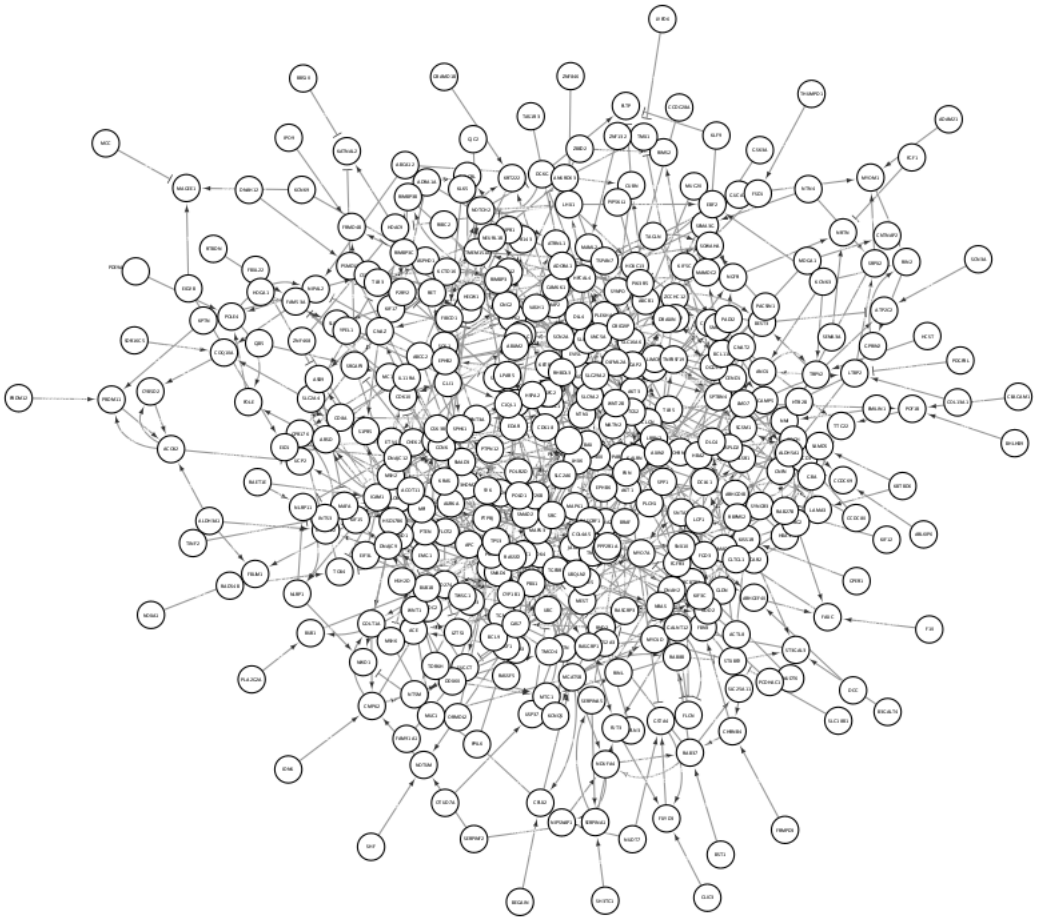


Figure 4.11: Consolidated fuzzy logic-based regulatory network

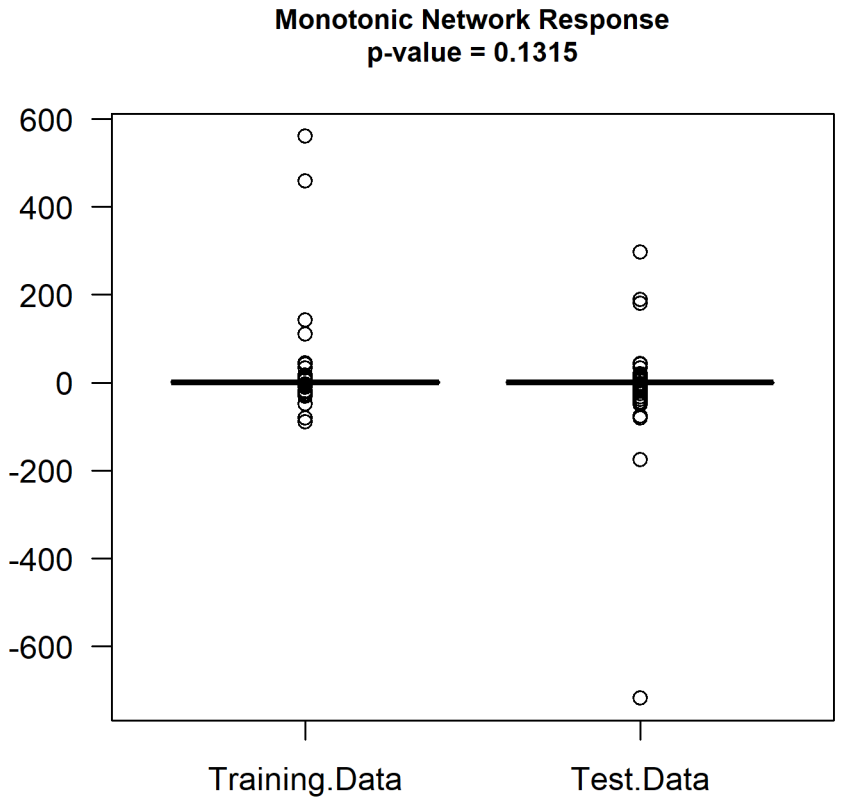


Figure 4.12: Distribution of observed monotonic changes for all network features over a $5,000$ time step dynamic simulation

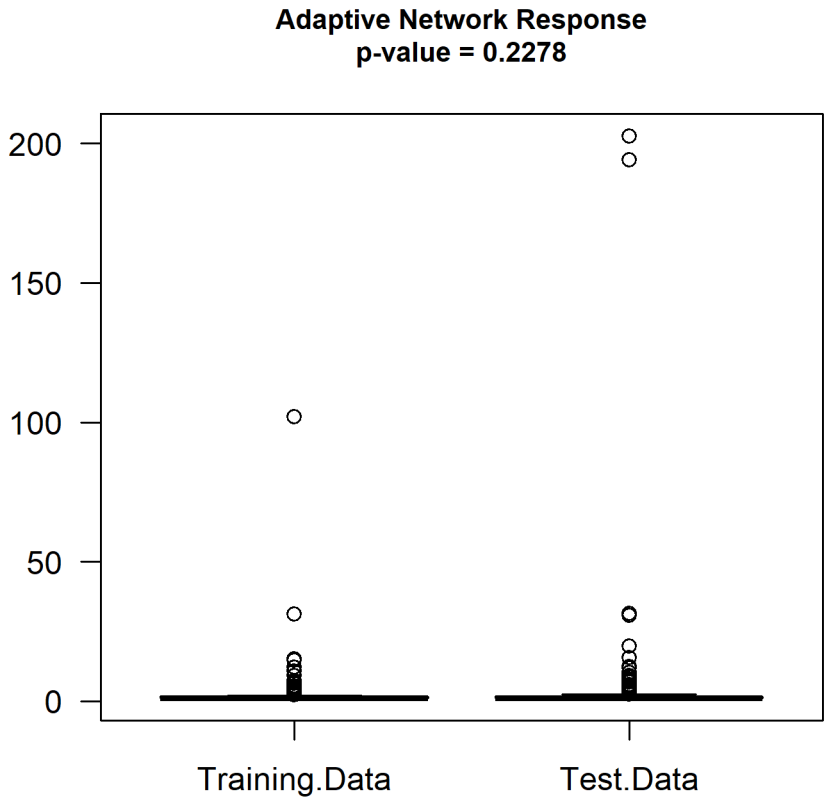
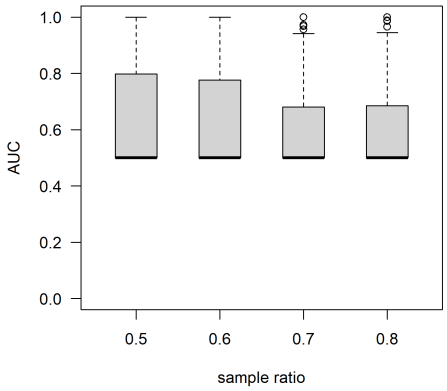


Figure 4.13: Distribution of observed adaptive changes for all network features over a $5,000$ time step dynamic simulation

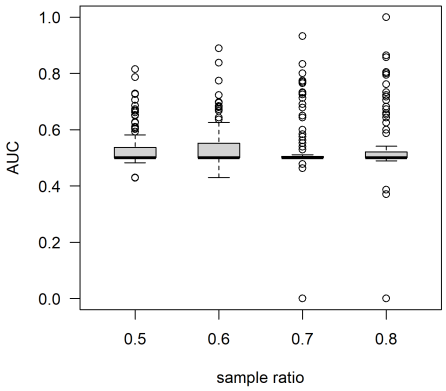
Table 4.12: Top 40 ranked regulatory features by node importance score estimates.

Gene Symbol	Description	Rank
UBC	ubiquitin C	1
PTEN	phosphatase and tensin homolog	2
SMAD2	SMAD family member 2	3
LMO7	LIM domain 7	4
GNAZ	G protein subunit alpha z	5
POLR2D	RNA polymerase II subunit D	6
TP53	tumor protein p53	7
AKT1	AKT serine/threonine kinase 1	8
RIMBP3	RIMS binding protein 3	9
CCNK	cyclin K	10
TNS1	tensin 1	11
PSMD13	proteasome 26S subunit, non-ATPase 13	12
PXN	paxillin	13
RIMBP3B	RIMS binding protein 3B	14
RIMBP3C	RIMS binding protein 3C	15
APC	APC regulator of WNT signaling pathway	16
GALNT12	polypeptide N-acetylgalactosaminyltransferase 12	17
MAPK1	mitogen-activated protein kinase 1	18
PARVG	parvin gamma	19
CTNNB1	catenin beta 1	20
MAPK3	mitogen-activated protein kinase 3	21
SMAD3	SMAD family member 3	22
MAGEE1	MAGE family member E1	23
WNT3A	Wnt family member 3A	24
RASGRP1	RAS guanyl releasing protein 1	25
SERPINA1	serpin family A member 1	26
GPR176	G protein-coupled receptor 176	27
TP53I3	tumor protein p53 inducible protein 3	28
SRC	SRC proto-oncogene, non-receptor tyrosine kinase	29
DDX60	DExD/H-box helicase 60	30
POLD1	DNA polymerase delta 1, catalytic subunit	31
POLE	DNA polymerase epsilon, catalytic subunit	32
GSK3B	glycogen synthase kinase 3 beta	33
TGFBR2	transforming growth factor beta receptor 2	34
KISS1R	KISS1 receptor	35
HSPG2	heparan sulfate proteoglycan 2	36
BCL9	BCL9 transcription coactivator	37
PTPRJ	protein tyrosine phosphatase receptor type J	38
MGAT5B	alpha-1,6-mannosylglycoprotein 6-beta-N-acetylglucosaminyltransferase B	39
BUB1B	BUB1 mitotic checkpoint serine/threonine kinase B	40



[t]0.5

Figure 4.14: Training data



[t]0.5

Figure 4.15: Test data

Figure 4.16: Distribution of AUC estimates in training and test data

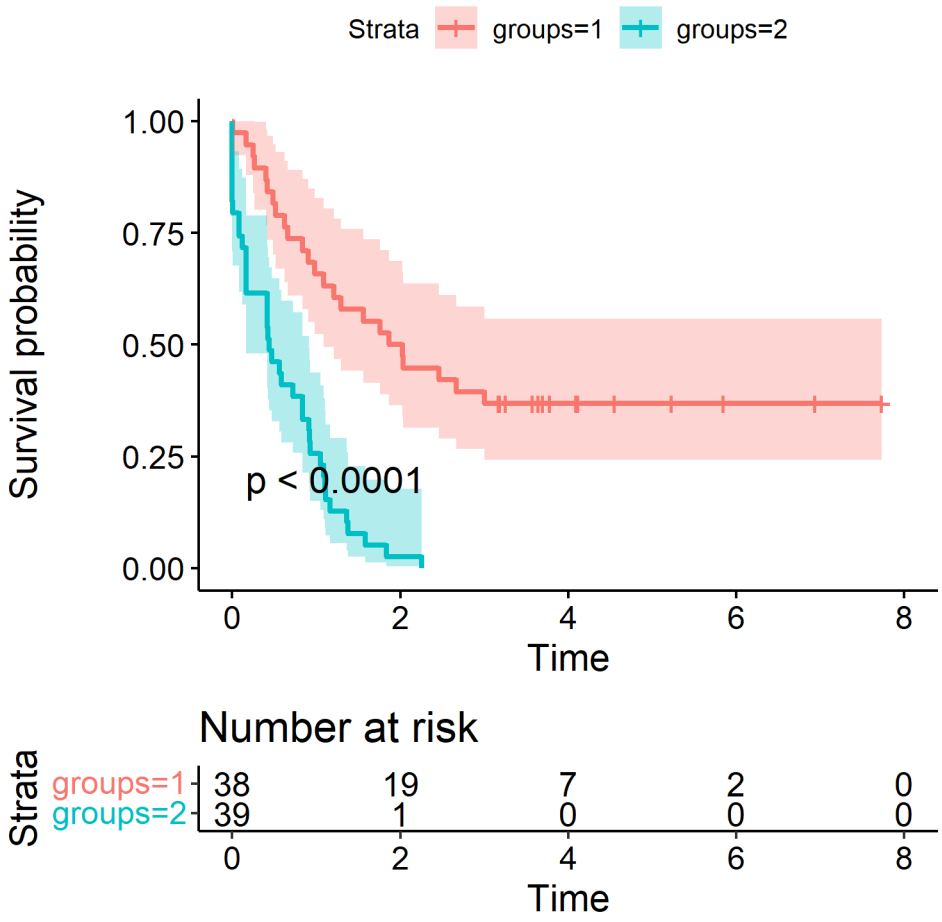


Figure 4.17: Kaplan-Meier (KM) curve of survival of patients. The groups 1 and 2 were identified by optimal separation of predicted responses from a Cox proportional hazard fit of the survival model

Chapter 5: Discussions and Conclusions

5.1 Discussions

This dissertation work has focused on elucidating the regulatory process in the synthetic lethal relationship between histone deacetylase and its synthetic lethal partners – the siRNA knocked down genes. Synthetic lethality is the phenomenon where the absence of the product of two genes, either naturally or artificially induced, selectively causes cellular death but individual deletion or absence of one of such does not. Emphasizing some of our study rationales discussed (chapters 1 and 2), this promises to facilitate a more rational and effective design of therapies directed at killing cancer cells. After a general overview of synthetic lethality and a background on fuzzy logic and approaches employed in our study (chapter 2), the materials and methods we employed were presented (chapter 3). In the previous chapter (chapter 4) we presented results, particularly related to the Fuzzy logic approach to infer mechanistic relationship of the processes that potentially converge on the observed phenotype in synthetic lethality, in the vorinostat-resistant (HCT-116) colon cancer cell lines. General discussions on inferred regulatory network and a note on the complexity of the employed fuzzy logic model is presented here, along with some conclusions.

5.1.1 Targeted therapy in colorectal cancer

Though the prevalence of colon cancer appear to be on the decline particularly in the above 65 year olds, the rising incidence in the younger population remain of significant concern. Besides surgical excision of tumor tissue and the use of classical chemotherapeutic regiments such as 5-FU, oxaliplatin, irinotecan, capecitabine, leucovorin, etc. with or without radiation, the search for more rational therapy continue to be of significance. The approval

of bevacizumab, panitumumab, and cetuximab for colorectal cancer supports the potential clinical benefit of rationally designed therapies – targeted therapies developed to take advantage of unique alterations specific to cancer cells to maximize the desired therapeutic effect while minimizing the toxicity in normal cells [313]. The currently approved targeted therapies for metastatic, stage IV or recurrent colorectal cancer include aflibercept, ramucirumab, panitumumab, regorafenib, and pembrolizumab. These are designed against the vascular endothelial growth factor (VEGF) or the epidermal growth factor receptor, both of which are tyrosine kinases. Targeting the VEGF pathway, bevacizumab and ramucirumab are developed as monoclonal antibodies while aflibercept, a recombinant fusion protein. Cetuximab and panitumumab, targets the EGFR pathway upstream of KRAS, and particularly effective in non-KRAS activating mutation cancers. Pambrolizumab is an antibody that targets the programmed cell death 1 (PD-1) protein in patients with microsatellite unstable tumors – associated with germline defects in the MLH1, MSH2, MSH6, and PMS2 genes[314, 315]. The proposed relationship of pambrolizumab to the DNA mismatch repair genes or gene products (MLH1, MSH2, MSH6, and PMS2) is less direct, compared to that of bevacizumab, ramucirumab or aflibercept to VEGF. Pambrolizumab targets the PD-1 protein on T cells, preventing its association with the PD-L1 ligand on tumor cells, macrophages or other tumor-infiltrating lymphocytes and myeloid cells acting in concert to suppress T-cells activation[316]. T cell activation results from presentation of mutation-associated neoantigen (MANA) that results from protein products resulting from DNA mismatches in complex with the major histocompatibility complex (MHC) protein [317].

5.1.2 Vorinostat, a form of targeted therapy

Analogous to pambrolizumab and these targeted therapies, vorinostat can act in both direct and indirect ways on the molecular pathways to regulate oncogenic processes. In addition to inhibiting histone deacetylases, evidences also point to its effect on the posttranslational modification state of proteins involved in oncogenesis and tumor supression. Vorinostat inhibits the removal of acetyl group from the ϵ-amino group of lysine residues of histone

proteins by histone deacetylases (HDACs) maintaining chromatin in an expanded state and thus facilitating transcriptional activities of major regulatory gene products such as transcription factors, cell-signaling regulatory proteins, and proteins regulating cell death[35]. Particularly described is its effect on the cyclin-dependent kinase inhibitor 1A, CDKN1A (also known as p21, WAF1/CIP1) gene transcription[36]. Richon et al found that vorinostat selectively induces CDKN1A expresion[36]. By binding to cyclin dependent kinases, CDKN1A prevents the phosphorylation of cyclin-dependent kinase substrates and thus block cell cycle progression[318]. Its non-histone-protein related effect include increased DNA binding of the transcriptional activator TP53 (Tumor protein 53) from increased acetylation, and a consequent increase in p53-regulated gene transcription rate. Its many diverse effect on the relationship of the TP53 gene and gene product may yet be relevant in the development of resistance or re-sensitization in colorectal cancer as evident from inferred interactions by our fuzzy logic approach (see Fig. 5.1). Also, BCL6 repression of transcription is inhibited by increased acetylation as a result of vorinostat[319]. As opposed to increased transcription, vorinostat (HDACi) represses the expression of genes cyclin D1, ErbB2, and thymidylate synthase [320]. Evidently, the effect of vorinostat tips the cellular equilibriun toward cell cycle arrest, anti-proliferation and apoptosis. It thus appeal to reason that molecular pathways of resistance would be quite the opposite. In attempts to circumvent resistance, Falkenberg et al had through a functional genomics screen identified genes that when knocked down by RNA interference (RNAi) sensitized cells to vorinostat-induced apoptosis. In other words, when these genes are knocked down, they co-operated with vorinostat to induce tumour cell apoptosis in otherwise resistant cells (synthetic lethality). These included – BEGAIN, CCNK, CDK10, DPPA5, EIF3L, GLI1, JAK2, NFYA, POLR2D, PSMD13, RGS18, SAP130, TGM5, and TOX4 (see Table 5.1)[39], all of which are pro-proliferative and potentially oncogenic. Of importance to our study however is to determine the molecular processes underneath the observed synthetic lethal phenotype and potential clinical significance. In a follow-up study, Falkenberg et al had validated the GLI1

gene as co-operative with vorinostat[40] to induce cell cycle arrest and apoptosis in otherwise vorinostat-resistant colon cancer cell lines. Given GLI1's role in the sonic hedgehog pathway, we had hypothesized that resistance to vorinostat is a result of uptick in embryonal gene regulatory programs. We also hypothesized that elucidated regulatory mechanism would include crosstalks that regulate this biological processes – embryonal gene regulatory programs.

Table 5.1: Table of identified synthetical lethal gene partners to histone deacetylase by Falkenberg et al.

	GENENAME
BEGAIN	brain enriched guanylate kinase associated
CCNK	cyclin K
CDK10	cyclin dependent kinase 10
DPPA5	developmental pluripotency associated 5
EIF3L	eukaryotic translation initiation factor 3 subunit L
GLI1	GLI family zinc finger 1
JAK2	Janus kinase 2
NFYA	nuclear transcription factor Y subunit alpha
POLR2D	RNA polymerase II subunit D
PSMD13	proteasome 26S subunit, non-ATPase 13
RGS18	regulator of G protein signaling 18
SAP130	Sin3A associated protein 130
TGM5	transglutaminase 5
TOX4	TOX high mobility group box family member 4

5.1.3 BCL2L1 downstream of GLI1

Glioma-Associated Oncogene Homolog 1 (GLI1) is a zinc finger protein and a transcription factor that acts downstream of the Hedgehog (Hh) signaling pathway. It mediates morphogenesis, cell proliferation and differentiation[321–325]. From reports of its potential relationship to GLI1 and vorinostat, Falkenberg and collagues reported the repression of the BCL2L1 gene on GLI1 knockdown. Bcl-2-Like Protein 1 (BCL2L1) is a cell death inhibitor, it inhibits the activation of caspases by binding to and blocking the voltage-dependent anion channel (VDAC), preventing the release of the caspase activator, CYC1, from the

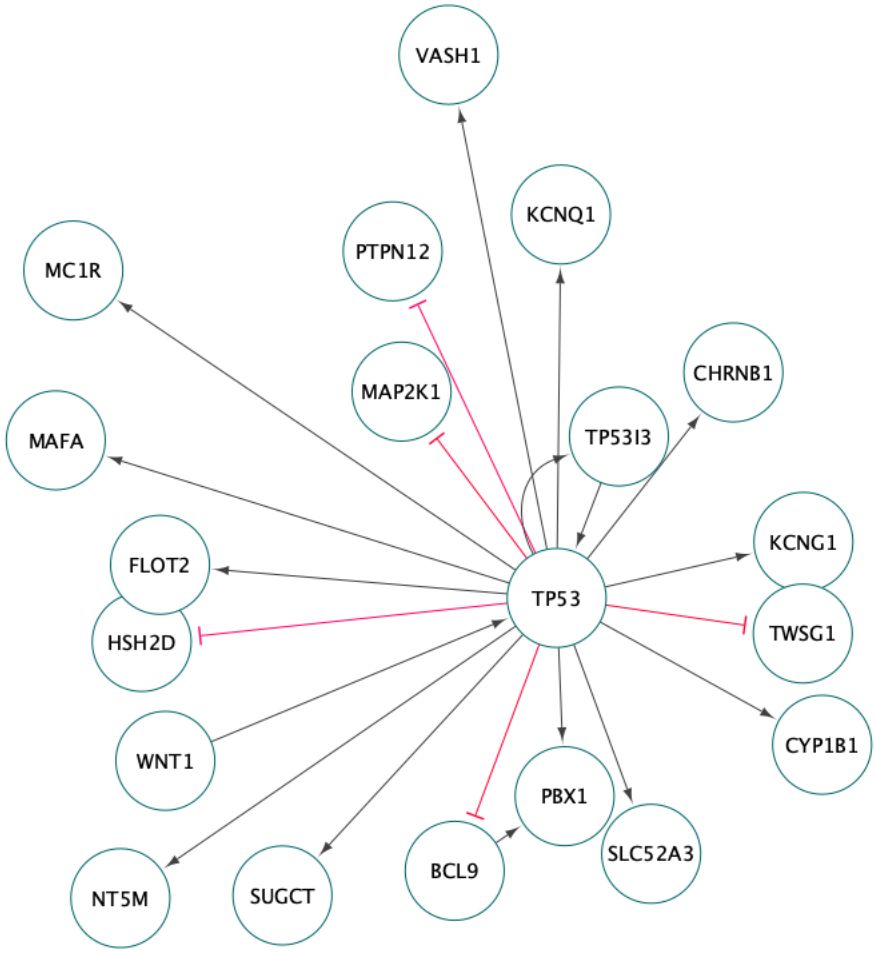


Figure 5.1: Fuzzy logic inferred TP53 molecular interactions. With a node degree of 20, the many interactions associated with the TP53 feature may support known consequent effect of vorinostat on TP53. The increased acetylation of the TP53 protein (a non-histone effect) as a result of histone deacetylase inhibition leads to increase in its DNA binding and increased transcription rate of its target genes.

mitochondrial membrane[326, 327]. At a adjusted p-value of < 0.05, we found the differential expression of BCL2L1 to significantly change in up to 8 siRNA knockdown (synthetic

lethality) experiments, including siGLI1 knockdown (Adjusted p-value = 1.7444e−21, Log2 fold change = −0.6694, St. error = 0.0676), compared to controlled experiments. However, subject to our selection criteria, BCL2L1 did not make the list of selected features for regulatory network inference. Although it may not be generalized in this study, there is evidence of the potential utility of BCL2L1 repression consequent to GLI1 knockdown as a path to restoring sensitivity to vorinostat in resistant colon cancer cell lines.

5.1.4 GLI1 is independent of the Sonic hedgehog (SHH) signaling

The Sonic Hedgehog (SHH) pathway upstream of GLI1 consists of PTCH1, SMO, and SUFU. Binding of the Hedgehog ligand to the cell surface receptor Patched (PTCH) releases its inhibitory effect on Smoothened (SMO), which in turn activates GLI1 [40, 328]. Suppressor Of Fused Homolog, SUFU down-regulates transactivation of target genes by GLI1 [329,330]. It forms a part of the co-repressor complex that acts on DNA-bound GLI1 and may act by linking GLI1 to BTRC – targeting GLI1 for degradation by proteasome [322,330–332]. Amongst TTRUSTv2 transcription factor-target database [333] retrieved 19 gene-targets of GLI1, only 2, AKT1[334] and SMAD4 [335] are contained in the derived fuzzy-logic regulatory network. Both of these are also known to be regulated by PIK3A and TGF-β respectively. Argawal et al. and, Nye et al. had suggested a form of cross-talk between the Sonic Hedgehog pathway and the cell proliferative pathways of AKT1 and TGF-β, mediated through GLI1 and GLI1-SMAD4 complex respectively. However, the non-significance of other members of the SHH pathway in our derived fuzzy logic regulatory network questions its role in vorinostat-resistance or re-sensitivity in the HCT116 colorectal cancer cell lines. The significance of other canonical upstream regulators of AKT1 and SMAD4, PIK3CA (node importance rank = 77; model fit = 0.701, model adjusted p-value = 0.00074) and TGFBR2 (node importance rank = 34, model fit = 0.704, model adjusted p-value = 0.00057) respectively suggests alternate mechanisms of resistance or re-sensitivity (Figs 5.2 and 5.3). These interactions may in part explain the proliferative and anti-proliferative processes observed in vorinostat resistant and re-sensitized colon cancer

cell lines independent of the SHH pathway. In some form multiple feed-forward (FF) and positive feed-back (PFB) control manner, GLI1 on the other hand appears to be under regulatory control with RET and ETV4, which themselves are tightly regulated by NEURL1B and AURKA. A consequently amplified GLI1 signal activates ABLIM2, whose signal is tempered by SRC. (Fig. 5.4). Using in-vivo cell culture and xenograft models, Ruan et al. recently showed that RET (rearranged during transfection) enhanced transcriptional activation by HH, independent of the SHH pathway. They showed that inhibition of GLI1 led to a reduction of RET-induced proliferation of SH-SY5Y cells and outgrowth of xenografts[336]. The role of GLI1 on RET expression in neuroblastoma is well documented – GLI1 induces the expression of RET[337, 338]. Zhu et al in a xenograft model, showed that EVS variant transcription factor 4 (ETV4) depletion inhibits the CXCR4/SHH/GLI1 signaling cascade in breast cancer[339]. Such may yet be the case in the vorinostat resistant colorectal cancer cell lines.

5.1.5 The PIK3CA-AKT1-ANO1 escape path

PIK3CA upregulation of anoctamin 1 (ANO1) through AKT1 may be a mechanism of resistance to circumvent vorinotat's activity, particularly in response to GLI1 knockdown. Mazzone et al [340], using a luciferase reporter system to determine ANO1 promoter activity, chromatin immunoprecipitation, siRNA knockdown, PCR, immunolabeling, and recordings of Ca^{2+}-activated Cl^- currents in human embryonic kidney 293 (HEK293) cells showed that binding of GLI1 represses ANO1 expression. They also showed that knocking down of GLI1 expression and inhibition of its activity increased the expression of ANO1 transcripts and Ca^{2+}-activated Cl^- currents in HEK293 cells. Relating to the activity of PIK3CA, Mroz and colleagues [341] showed that induction of the transmembrane protein 16A (TMEM16A) also known as ANO1 expression is mediated by a sequential activation of phosphatidylinositol 3-kinase (PIK3) and protein kinase C-δ (PKCδ). Our fuzzy logic approach indicates an

involvement of AKT1 (Fig. 5.2). We suppose that in response to vorinostat, the PIK3CA-AKT1-ANO1 pathway provides alternate escape pathway from anti-cell profliferation signatures.

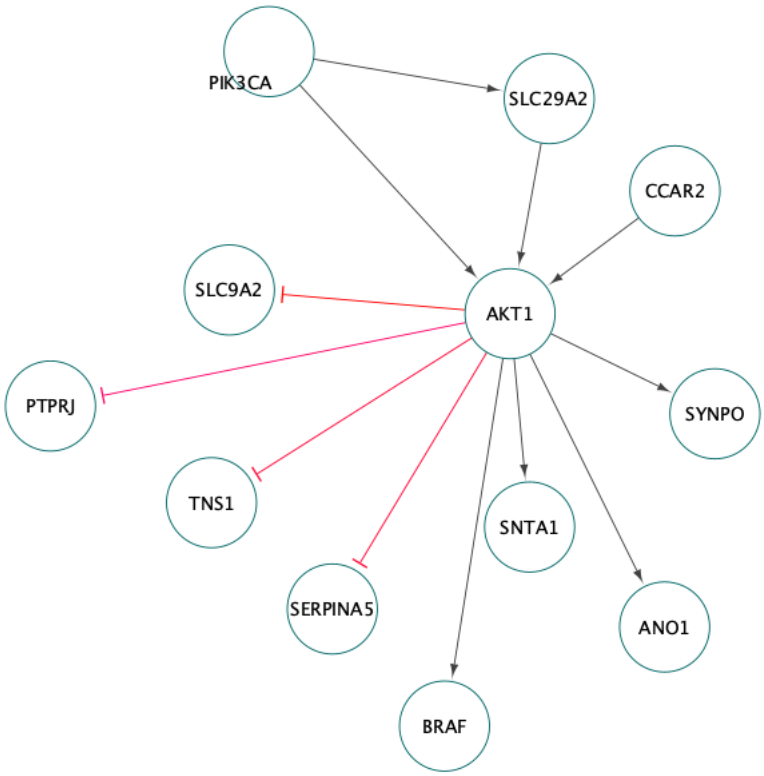


Figure 5.2: The AKT1 Pathway. The PIK3CA-AKT1-BRAF relationship remains consistent as with canonical cell pro-survival and proliferation pathway. Besides less known activation pathways involving dowstream activation of SNTA1, ANO1 and SYNPO, the canonical AKT1 activation of BRAF is highlighted by the fuzzy logic inference method. PIK3CA upregulation of ANO1 through AKT1 may be a mechanism of resistance to circumvent vorinotat's activity, particularly in response to GLI1-knockdown (see text).

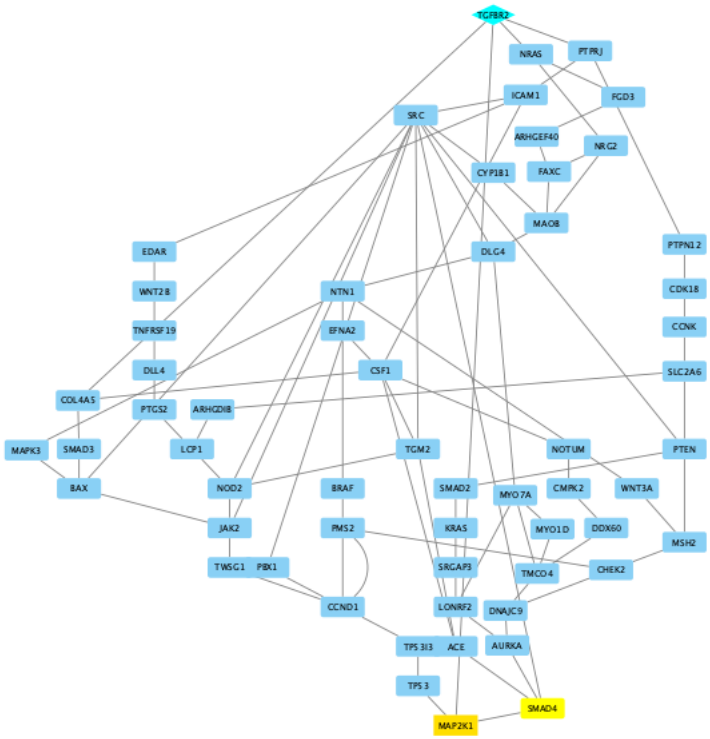


Figure 5.3: The TGFBR2-SMAD4 subnetwork. The Fuzzy logic based network inference approach shows canonical and non-canonical interactions that connect the TGFBR2 to SMAD4. Canonically, activated carboxy-terminal phosphorylated SMADs (SMAD2 and SMAD3) partner with their common signal transducer SMAD4 and translocate into the nucleus to regulate diverse biological activites, mostly by partnering with transcription factors. Inferred fuzzy-logic regulatory network includes both direct and indirect relationships amongst gene and gene-products related to the SMAD signaling complex.

5.1.6 The pro-survival and anti-proliferation balancing act

With evidences pointing towards activation of cell pro-survival and cell proliferation pathways independent of SHH, the role of these pro-survival and cell proliferation pathway members (PIK3CA, AKT1, MAPK1, MAPK3, WNT3A etc) in vorinostat resistance or vorinostat sensitivity in colorectal cancer are worth evaluating. Interestingly observed are

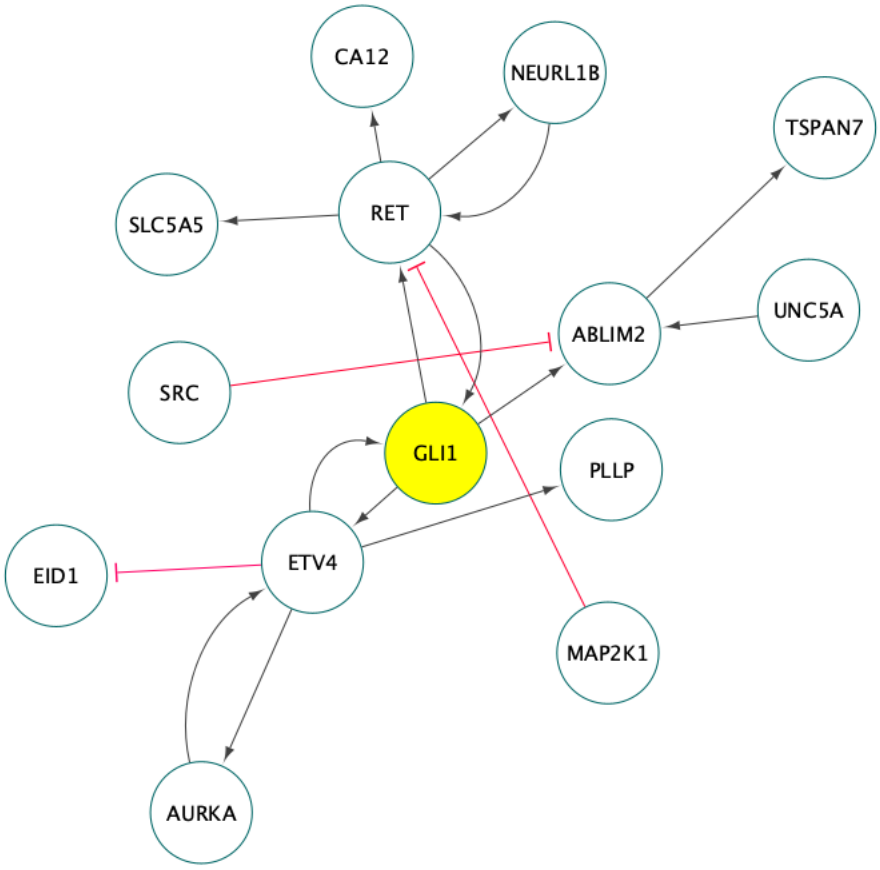


Figure 5.4: Inferred GLI1 interactions based on fuzzy logic. GLI1 activation or regualation appears to be independent of members of the Sonic Hedgehog (SHH) pathway. In some form multiple feed-forward (FF) and positive feed-back (PFB) control manner, GLI1 appears to be under regulatory control with RET and ETV4, which themselves are tightly regulated by NEURL1B and AURKA respectively. Amplified GLI1 signal activates ABLIM2, whose signal is tempered by SRC.

the almost equal or more significant representation of tumor suppressor genes (PTEN, TP53, APC, UBC, GSK3B, etc) top-ranked in terms of node importance in the derived regulatory

network. It appears resistance or sensitivity is a balancing act between pro-survival and anti-proliferation signatures – relationships thar appear to be clinically significant, given top-ranked features' expression being predictive of colorectal cancer patients survival (Fig. 4.17, log rank, p-value < 0.0001) in the sampled population. Here is presented a rationale for including anti-cell prosurvival and anti-cell proliferative genes' targeted therapy, in combination with vorinostat therapy to improve patient survival in colorectal cancer.

5.1.7 Complexity of the Fuzzy Logic Model

Appealing to natural reasoning, the simplicity of the fuzzy logic approach may be deceptive. The attending rates of growth is exponential [342,343] and tends toward a combinatorial explosive rate, particularly in modeling higher order regulatory elements – a fact not quickly apparent. This is particularly so in inferential problems with more than the three state (LOW, MED, and HIGH) and three inter-actor fuzzy set model. Aware of this, many prior complexity models have expressed concerns and have limited their search space for molecular inter-actors to triplet models (that is, models with only two regulators and one output feature) [342, 344]. However, with increasing desire to model higher order interactions or perform an exhaustive search within the attending search space, the time complexity of fuzzy logic algorithms very rapidly grows, with a resultant need for alternative approaches. In evaluating time complexity, our empirical analysis and results show that models based on just the exponential growth function component of complexity underestimates time requirement.

Much have been described with respect to exponential growth while less have been said of the combinatorial growth rate resulting from increase in feature space, n. Although with a lesser recognition, the likelihood of combinatorial explosion in potential regulatory networks that can be explained by a few number of genes have been documented. Edward and Glass (2000) described a combinatorial solution to the question of how many distinct logical structures exist for n-dimensional networks. They showed that the number increases very rapidly with n [345]. With our observation that computational time complexity analysis

based off only the exponential growth function underestimates time requirement, it would stand to reason to consider this component, n, and a resultant combinatorial growth rate in estimating time complexity as well. Al Qzlan and colleagues in their review of fuzzy methods' state of the art, acknowledged the potential attendant complexity associated with modeling complex regulatory networks using the fuzzy logic approach, suggesting implementation time could be on the scale of years instead of hours [346]. This notion may not be far from the truth without an optimization and a re-think of approaches to implement the fuzzy logic model. This dissertation sought to model a complex regulatory network that would almost be impractical within a reasonable time frame going by Al Qazlan and colleagues' notion. However, our our hyperparallel optimization approach (see Appendix II) is a step towards a more efficient utility of the fuzzy inference system on biological data.

The *multi-staged hyper parallel* approach

To address this bottleneck – personally described as *rate-limiting*, we developed the *multi-staged hyper parallel* approach. The *multi-staged hyper parallel* approach is a form of the divide-and-conquer type computational algorithm, but not exactly the same. With typical divide-and-conquer algorithms, increased computational resource utility comes with increased cost of work done in managing the resources. This increase in cost, which is limiting in nature, results in a non-linear growth of achievable speedup and causes maximally attainable speedup to approach an asymptotic maximum, as observed in this study. However with our multistaged hyperparallel algorithm, a solution is designed as such to follow a Gustafson's speedup prediction model instead of the Amdahl's model. And, with increasing computational resources, the approach is able to attain even higher orders of magnitude with a significant improvement to computational time complexity of model inference.

5.2 Conclusions

Circumventing complex and sensitive hyperparameter estimation, the Fuzzy logic model appeared to offer an easily comprehensible and in theory, a highly-scalable approach to

extract mechanistic explanations from high-throughput biological data. In this study each siRNA knockdown experiment was considered a transition state toward a unitary state of cellular death in colon cancer cell lines. To elucidate the processes that converge on the synthetic lethal state so as to facilitate a more rational therapeutics design, we employed the fuzzy logic approach. We identified direct and indirect regulators of sensitivity or resistance in the vorinostat-resistant (HCT116) cell lines. We validated inferred models in an independent dataset. We identified direct and indirect regulators of resistance or re-sensitivity i.e. our knowledge-guided fuzzy logic approach is able to tease the regulatory mechanism involved in histone deacetylase inhibition resistance in colon cancer cell lines from the biological dataset. We had hypothesized a resistant mechanism that likely involved the Sonic Hedgehog pathway – embryonal gene regulatory pathway.

However from our study, there is no significant evidence that vorinostat resistance is due to an upregulation of embryonal gene regulatory pathways. Our observation rather support a topological rewiring toward canonical oncogenic (pro-cell survival, cell proliferative) pathways, including the PIK3CA, AKT1, RAS/BRAF etc. pathways. Exploring the potential clinical or biomedical significance, our inferred major regulatory molecules are able to delineate patients into high- and low- risk of mortality. The identified key regulatory network genes' expression profile are able to predict short- to medium-term survival in colorectal cancer patients – providing a rationale for an effective combination of therapeutics that target these genes (particularly for the pro-cell survival and cell proliferative gene products) along with vorinostat in the treatment of colorectal cancer.

Chapter 6: Future Directions

The Fuzzy logic mechanistic model in elucidating molecular regulation is very powerful and appealing. Not only for its simplicity and ease of interpretation in rational linguistic terms that are readily understood by a lay audience, but appealing for its ease with incorporating concepts from other qualitative and quantitative fields. And, it does find application in many diverse fields of biomedical investigation and inquiry. With respect to the dissertation work presented here, future possible directions include, but not limited to the following:

6.1 Improving Computation-time Complexity

6.1.1 Extending beyond the boundaries of achieved speed-up

As it has been noted in this dissertation, that the computational complexity of the fuzzy logic regulatory model, particularly at higher order of interactions, quickly approaches those of more complicated models. Approaching computationally intractable problems, the benefit inherent in the simplicity and strength of fuzzy logic models near being undermined. The multi-staged hyperparallel optimization method presented in this dissertation represents one of other potentially possible approaches that seeks to push the boundary of that which is attainable in terms of clock-speed and model search space. With advances in computing technologies and available resources, this is anticipated to get better. An immediate short-term future direction may be to enable GPU (Graphics Processing Unit) compute capabilities in the implemented fuzzy inference engine to offset the cost of communication between hundreds to thousands of compute nodes [347–351].

6.2 Hybrid Fuzzy Logic Models

Moving a bit past the exhaustive search paradigm as employed in this dissertation work, a future direction may also be to explore the performance of hybrid approaches in elucidating verifiable models of synthetic lethality. As mentioned earlier, an appeal of the fuzzy logic model is its ease with incorporating concepts from other qualitative and quantitative approaches. Hybrid approaches being considered include the Fuzzy Cognitive Maps, Neuro Fuzzy, and the Fuzzy Petri Nets. Fuzzy Cognitive maps (FCMs), combine features from fuzzy logic and Artificial Neural Networks [352,353]. The Neuro-Fuzzy approaches are somewhat similar but not exactly the same as the Fuzzy Cognitive Maps (FCMs)[354, 355] [356].

6.3 Multi-component Fuzzy Models

This dissertation has focussed on retrieving models from RNA sequencing gene expression data. However, RNA sequencing quantifications are only surrogates to true protein expression. Also, beyond RNA and proteins, other macromolecules or bio-molecules play different roles in real-life true models of physiological or pathological processes. These interacting biomolecules can be modeled as nodes in a regulatory network whose edges represent regulatory or metabolic relationships [357]. How these all converge to explain synthetic lethal states and how the knowledge of these better refine our therapeutic intervention, which remains to more effectively and selectively kill cancer would be worth pursuing.

Appendix A:
jFuzzyMachine – A Fuzzy Logic-based Inference Engine for Biological High-throughput Data

A.1 Introduction

In spite of advances in the theoretical basis and relative biological validity of the fuzzy approach, there exists the very apparent lack of analytical tools [358] that implement these methods available to the scientific and research community (Table A.1). The apparent lack of readily available community tools limit the applicability and benefits of the fuzzy inference system to biological data. This also limits necessary comparisons and benchmarking of results obtained by the method to those obtained from comparable methods. To elucidate mechanistic relationships from generated high-throughput biological data, and to address the aforementioned gap, we developed the jFuzzyMachine – a freely available fuzzy logic based inference engine for biological data.

Table A.1: Fuzzy logic-based regulatory inference tools availability. Combined fuzzy clustering and Bayesian networks (FCBN), Fuzzy cognitive map (FCM), Fuzzy Petri net (FPN), Ant Colony Optimization (ACO), Activator-Repressor Regulatory Model (ARRM)

Modeling method	References	Tool Available for Free
Fuzzy logic + clustering	[359]	No
Fuzzy rules	[360]	No
Fuzzy logic + clustering	[361]	No
AFEGRN	[362]	No
Coalesce GRN (CGRN)	[363][364]	No
FRBPN	[365]	No
FCBN	[366]	No
ODEs + FIS	[367]	No
FCM + clustering	[368]	No
FPN	[369]	No
FCMs + ACO	[370]	No
ARRM + SRS	[371]	No
Fuzzy Mining Model	[372]	No

A.2 Design and Implementation

The jFuzzyMachine tool is implemented in the platform-independent Java programming language to facilitate an extensive community reach. It is modular in design to facilitate an easy decoupling of component parts. It consists of: 1) the Initiation Module, 2) the Main Module, and 3) the Utilities Module (see Figure 1). The 'Initiation Module' consists essentially of the program configuration and run parameter specification units. Depending on user-desired added-functionality beyond regulatory model inference, a user may choose to specify parameters that apply only to desired post-inference processing. The Main Module houses the application's main functionality – the fuzzy logic based regulatory inference engine. The module implements the: fuzzification, rule evaluation, and defuzzification schemes [360,361][373]. It currently implements the 'Union Rule Configuration' (URC) rule evaluation scheme of Coomb's et al [374,375] and an optimized version of the 'Exhaustive search' algorithm of Sokhansaj et al [376]. The Utilities Module consists of two submodules: a) the Postprocessing Submodule and, b) the Add-ons (or Plug-ins) Submodule. The Postprocessing Submodule consists of three Units – the 'Graph', 'Evaluation (or Validation)', and the 'Dynamic Simulation' Units. The Graph Unit consolidates the best fitted models derived from the fuzzy inference system into a network graph as explained in Gormley et al [377]. The Evaluation Unit simply compares expression profile predictions by inferred models to the experiment observed values. Depending on the user-specification, this may be against original model elucidation data (default) or an independent dataset. Also depending on the user, a re-calculation of the model's fit may be specified, particularly to quantitatively describe how well models fit independent datasets. The 'Dynamic Simulation Unit' implements and executes model dynamic simulations as also described in Gormley et al [377]. To facilitate downstream data integration, the dynamic simulations' stop criteria is dependent on the user-specified number of iteration steps and not the computed error. However the error estimates at the end of the simulation runs are reported. In anticipation

of community contributions, the Add-on (or Plug-ins) Submodule is described. Current in-house created functionalities that would fit appropriately configured add-on units include an "In-Silico Knockout Simulations" add-on and a "Visualization" add-on which depends on a secondary-installed program. These are also freely available on request.

A.3 Demonstration

We fully demonstrate current functionalities of the jFuzzyMachine tool in [378] and [379]. Please see or request these publications.

A.3.1 Getting jFuzzyMachine

jFuzzyMachine's source codes and precompiled binaries may be requested or freely downloaded from the bitbucket git repository locations:

https://bitbucket.org/paiyetan/jfuzzymachine/src/master/ and
https://bitbucket.org/paiyetan/jfuzzymachine/downloads/.

The application and distributed binaries are made available in a compressed folder named jFuzzyMachine.zip.

A.3.2 Installation Requirements

jFuzzyMachine is platform independent. It would run on a Windows, Mac, or UNIX-based Operating System (OS) with an appropriately pre-installed Java Runtime Environment (JRE). Java 7 or above is required. You may download the latest version of Java from https://www.java.com/en/download/.

To run the visualization add-on (plugin), provided as an added-value, a UNIX-based OS with the R program statistical computing environment [269] pre-installed, is required. R may be downloaded from https://cran.r-project.org/.

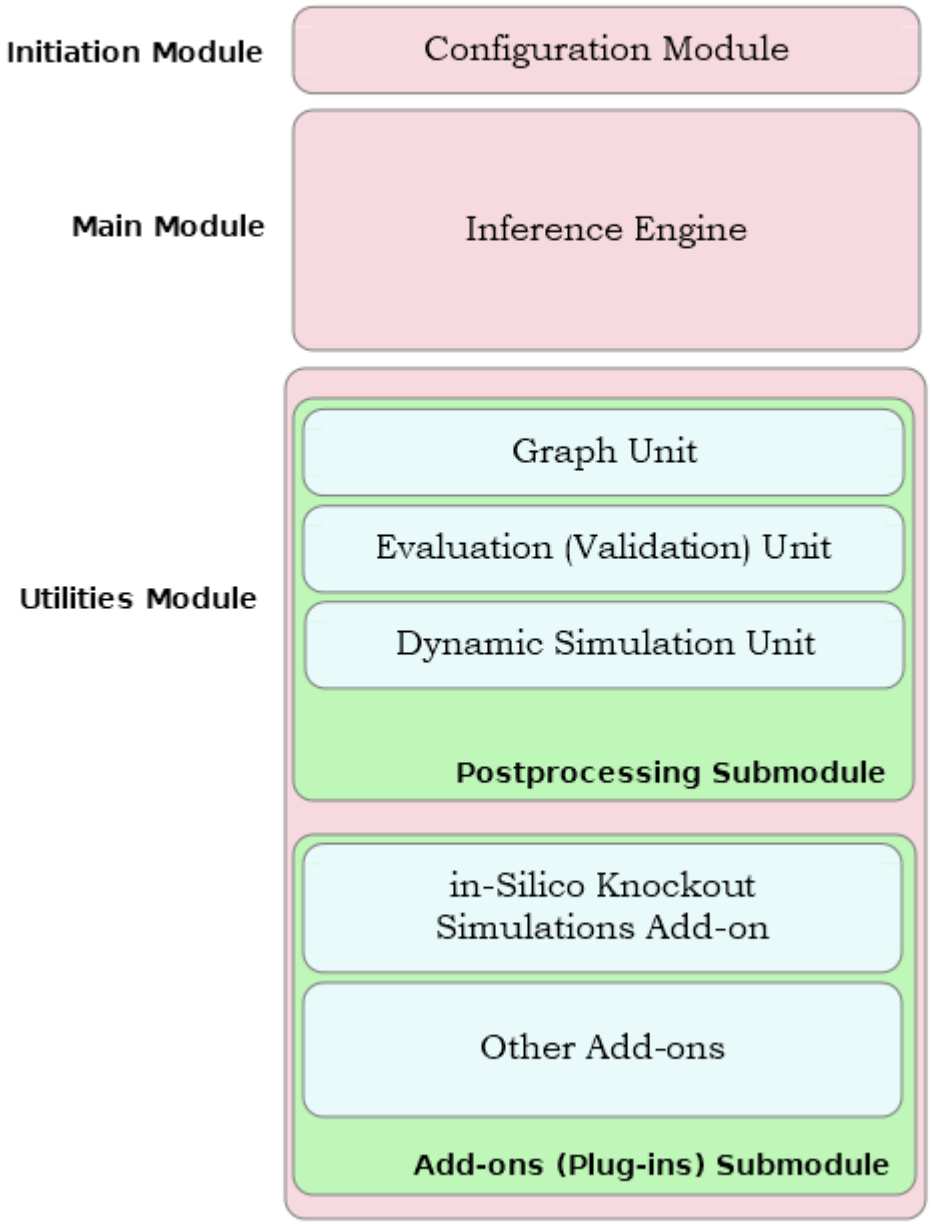


Figure A.1: The jFuzzyMachine Application Components.

A.3.3 Installing jFuzzyMachine

Unzip the compressed application package into a directory of choice. The content of the unzipped folder should include: One primary java archive (.jar) folder, four runtime configuration (.config) files, and four subdirectories (etc/, lib/, plugins/, and src/),

- JFuzzyMachine.jar
- jfuzzymachine.config
- jfuzzymachine.graph.config
- jfuzzymachine.evaluator.config
- jfuzzymachine.simulator.config
- etc
- lib
- plugins
- src

The configuration files are pre-filled to satisfy required parameters for this manual's demonstration. Users may appropriately fill-in their own specifications and experiment with the tool. See configuration options below.

A.3.4 Running jFuzzyMachine

To run the tool, on the command-line,

1. Navigate into the application directory
2. Appropriately fill-in the desired run-time options in the configuration files and

3. Depending on application module or functional unit of interest, type the following commands, one at a time:

   To elucidate fuzzy logic-based regulatory relationships, run the commands

```
1        java -Xmx10G -cp JFuzzyMachine.jar jfuzzymachine.JFuzzyMachine
   \
2                    jfuzzymachine.config
3
```

   To derive a composite network graph, including rule frequencies, run

```
1        java -Xmx10G -cp JFuzzyMachine.jar jfuzzymachine.utilities.
   graph.Graph \
2                    jfuzzymachine.graph.config
3
```

   To evaluate or validate how well inferred fuzzy logic-based regulatory models fit the data, run

```
1        java -Xmx10G -cp JFuzzyMachine.jar jfuzzymachine.utilities.
   ModelValidator \
2                    jfuzzymachine.evaluator.config
3
```

   To run dynamic simulation of regulatory network, and tease expression values at systems steady state, run

```
1        java -Xmx10G -cp JFuzzyMachine.jar \
2         jfuzzymachine.utilities.simulation.Simulator jfuzzymachine.
   simulator.config
3
```

The jfuzzymachine.config file

```
## jfuzzymachine.config
inputFile=./etc/projects/demo/inputs/exprsMat.txt
outputDir=./etc/projects/demo/outputs/runJFuzzy
maxNumberOfInputs=-1
numberOfInputs=3
outputInRealtime=TRUE
eCutOff=0.6
useAllGenesAsOutput=FALSE
iGeneStart=1
iGeneEnd=14
useParallel=TRUE

```

The jfuzzymachine.config has, at least, the above listed parameters ('key'='value' pairs). The associated values listed here are for demonstration purposes in this manual. The inputFile option specifies the relative path to the data matrix of normalized expression values. The outputDir specifies the path to the directory where results from jFuzyMachine are to be placed. The maxNumberOfInputs option is a flag that specifies how jFuzzyMachine should handle input (regulatory) features. A negative flag indicates that exactly the specified numberOfInputs option be considered. A positive value specifies to jFuzzyMachine to consider all possible number of inputs up-to the specified value. E.g. a positive value of 4, simply says to jFuzzyMachine to consider all possible combinations of $1, 2, 3,$ and 4 regulatory inputs to an output feature. Current implementation of jFuzzyMachine allows up to 5 inputs. A negative flag however, says to jFuzzyMachine to consider only possible combinations of 3 regulatory inputs (specified by the numberOfInputs option in the above configuration) to an output feature. The outputInRealtime option tells jFuzzyMachine to output its runtime informations onto a standard output (the console). This typically includes derived models, inferred regulatory rules, and computed fit estimates. The eCutOff option specifies the

cut-off for which to consider a computed fuzzy logic model. Models below the specified value are discarded. The useAllGenesAsOutputs option specifies whether to consider all features in the expression values matrix or a limited set specified by the iGeneStart and iGeneEnd options. The iGeneStart and iGeneEnd options specify the range of features to use from a numerically ordered list – the expression matrix row numbers. The options iGeneStart=1 and iGeneEnd=14 in the configuration above, simply says to jFuzzyMachine to consider features of expression profiles from the first to the 14^{th} row, in the expression matrix (inputFile), as probable outputs in the regulatory model inference. The useParallel option indicates whether to run jFuzzyMachine in the optimized mode for speed (distributed across computing cores available at runtime).

The jfuzzymachine.graph.config file

```
## jfuzzymachine.graph.config
exprsFile=./etc/projects/demo/inputs/exprsMat.txt
input=./etc/projects/demo/outputs/runJFuzzy
runId=_demo
fitCutOff=0.6
useAnnotatedGraphModel=TRUE
outputEdges=TRUE
topFittedModelsToOutput=150

```

The above are parameters (runtime options) to the jFuzzyMachine Graphical unit. The exprsFile option specifies a path to the expression matrix from which regulatory models were inferred. The input option specifies a path to the directory in which jFuzzyMachine inferred models and output result files are placed. The runId option is a user-specified identifier prepended to outputted results' filenames (please see the Results section). The nomenclature (naming convention) of the outputted result files is of the form

<runId>_runJFuzzUtils.<fileType>. The fitCutOff option specifies a cut-off for considering models. Models above specified fitCutOff are considered for inclusion in a consolidated network model. Only the regulatory edges of the passing models are considered. The useAnnotatedGraphModel tells jFuzzyMachine Graph unit to model regulatory network as a directed acyclic graph. If TRUE, the outputted adjacency matrix (.adj or .mat file) is a directed graph. By default, jFuzzyMachine's Graphical unit outputs only an adjacency matrix file, to represent the inferred regulatory network, but the option outputEdges specifies to jFuzzyMachine to also print edges (a .edg file) of the consolidated network. The topFittedModelsToOutput applies to ouputting fitted models. It specifies the number of alternate top ranked models that pass fitCutOff filter option to report in the output (.fit2) file.

The jfuzzymachine.evaluator.config file

```
1  ## jfuzzymachine.evaluator.config
2  exprsToValidate=./etc/projects/demo/inputs/exprsMat.txt
3  fitFile=./etc/projects/demo/outputs/runJFuzzy/runJFuzzUtils/
     _demo_runJFuzzUtils.fit
4  fitCutOff=0.6
5  validationType=validations
6
```

In jfuzzymachine.evaluator.config file, the exprsToValidate option specifies a path to the expression matrix from which regulatory models were inferred – in the case of evaluating the performance of the inferred models against the model–generating data. For an independent evaluation of the model, this is a path to the expression matrix of the independent dataset. The fitFile specifies the path to derived .fit file from jFuzzyMachine's Graph unit. The .fit file contains the best fitted models. The fitCutOff specifies an estimated fit cut–off value above which to consider models. The validationType option, specifies what validation is being performed. Acceptable values include validations (default) and ivalidations. Specifying ivalidations implies an independent validation is being performed.

The jfuzzymachine.simulator.config file

```
## jfuzzymachine.simulator.config
exprsMatFile=./etc/projects/demo/inputs/exprsMat.txt
edgesFile=./etc/projects/demo/outputs/runJFuzzy/runJFuzzUtils/
  _demo_runJFuzzUtils.edg
fitFile=./etc/projects/demo/outputs/runJFuzzy/runJFuzzUtils/
  _demo_runJFuzzUtils.fit
fitCutOff=0.6
simulationType=simulations
maxIterations=5000
eCutOff=0.000001
initialOutputsValues=ALL
alpha=0.01

```

In the jfuzzymachine.simulator.config, the exprsMatFile specifies a path to the expression matrix from which regulatory models were inferred – in the case of evaluating the performance of the inferred models against the model–generating data. For an independent evaluation of the model, this is a path to the expression matrix of the independent dataset. The edgesFile and fitFile options specify the path to the .edg and .fit files derived from jFuzzyMachine's Graph unit. These contain the edges of the consolidated network and the best fitted models from the regulatory model elucidating steps. The fitCutOff specifies an estimated fit cut-off value above which to consider models. The simulationType option specifies the sort of dynamic simulation to be performed. Acceptable values include simulations (default) and isimulations. Specifying isimulations implies a dynamic simulation of consolidated network, using previously derived models as simulation parameters on an independent dataset is being performed. The maxIterations, eCutOff, initialOutputsValues and alpha are other dynamic simulation parameters. The maxIteration and eCutOff are the stopping criteria – maximum iteration steps and error estimate cut–off respectively. Default values are 5000 and $10e-7$ respectively. To better facilitate integration with downstream

analyses, the Dynamic Simulation Unit defaults to preferably using the maxIteration option as stopping criteria. The initialOutputsValues specifies which 'perturbation', 'sample', or 'time–point' values, from the expression matrix, to use as initial values in the simulation. It defaults to ALL, i.e. all values are sequentially used. Other values are FIRST and RANDOM, implying the first column and a random column values respectively. The alpha option specifies the 'mixing parameter', α, of the simulation model. Based on Gormley et al [182], linear combination of new and old values ensures that the system smoothly converges towards equilibrium. And accordingly, jFuzzyMachine's Dynamic Simulation Unit computes new values of each node (I_1) based on the initial conditions and the fuzzy relations inferred from the data; values in the next iteration were calculated as a linear combination of the inferred values (I_n) and the initial values (I_{n-1}) as follows:

$$I_{n+1} = \alpha I_n + (1\alpha)\, I_{n1} \tag{A.1}$$

A.3.5 Results

The Main Module - Inference Engine

The main output results from the jFuzzyMachine inference engine are written to the outputDir. These are files or single file ending with .jfuz. From the demo run, this would be the ./etc/projects/demo/outputs/runJFuzzy/exprsMat.1.14.3.TRUE.jfuz. This consists of 4 major sections indicated by the > character at the begining of the line. These sections include; a prologue, run parameter listing, the main result, and an epilogue. The prologue section stores information such as the run's start-time, while the epilogue stores the run end-time and duration. The main section is a tab-delimited table with columns: Output, NumberOfInput(s), Input(s), Rule(s), and Error(E). The Output column indicates the output node in the model; the NumberOfInput indicate the number of input nodes, the Input(s), considered. The Rule(s) column indicate the fuzzy logic rule that associates the respective input node to the output node. The Error(E) column indicates the model's fit. Shown below is a sample output from the demo run:

```
> StartTime: Mon Jul 20 23:47:25 EDT 2020
> Search Parameters:
          inputFile = ./etc/projects/demo/inputs/exprsMat.txt
  maxNumberOfInputs = -1
     numberOfInputs = 3
   outputInRealtime = TRUE
            eCutOff = 0.6
useAllGenesAsOutput = FALSE
         iGeneStart = 1
           iGeneEnd = 14
        useParallel = TRUE
         outputFile = ./etc/projects/demo/outputs/runJFuzzy/exprsMat.1.14.3.TRUE.jfuz
     modelPhenotype = FALSE

Initiating...
Searching (Exhaustive Search)...
                 All Genes#: 14
   Output Nodes Considered#: 14
> Begin Search Result Table:
Output NumberOfInput(s) Input(s) Rule(s) Error(E)
PTHLH 3 [KRT86, RUBCNL, CYS1] [[1, 3, 3], [1, 3, 2], [2, 1, 1]]
    0.6057812852061071
PTHLH 3 [LINC00707, RUBCNL, LINC00634] [[1, 1, 1], [1, 2, 2], [2, 1, 1]]
    0.6395610849445412
PTHLH 3 [LINC00707, RUBCNL, LINC00634] [[1, 1, 1], [1, 2, 3], [1, 1, 1]]
    0.6055562465480613
PTHLH 3 [LINC00707, RUBCNL, LINC00634] [[1, 1, 1], [1, 3, 2], [2, 1, 1]]
    0.7444440833291
PTHLH 3 [LINC00707, RUBCNL, LINC00634] [[1, 1, 1], [3, 2, 3], [2, 1, 1]]
    0.6035443648873104
```

```
PTHLH 3 [LINC00707, RUBCNL, LINC00634] [[2, 1, 1], [1, 2, 2], [2, 1, 1]]
    0.664214821186895
PTHLH 3 [LINC00707, RUBCNL, LINC00634] [[2, 1, 1], [1, 3, 2], [1, 1, 1]]
    0.6474161595899811
PTHLH 3 [LINC00707, RUBCNL, LINC00634] [[2, 1, 1], [1, 3, 2], [2, 1, 1]]
    0.7321145800582836
  ...
C2orf78 3 [SERPINB7, CYS1, LINC00886] [[3, 3, 1], [1, 2, 3], [1, 3, 3]]
    0.6316025202379136
LINC00634 3 [LINC00886, GCNT4, MGAM] [[1, 3, 3], [3, 2, 1], [2, 1, 1]]
    0.6000086433874239
LINC00634 3 [KRT86, LINC00886, GCNT4] [[3, 1, 1], [2, 2, 3], [3, 1, 1]]
    0.6009588086189968
> End Search Result Table

...Done!
> Epilogue

   Started: 1595303245550: Mon Jul 20 23:47:25 EDT 2020
     Ended: 1595303288832: Mon Jul 20 23:48:08 EDT 2020
Total time: 43282 milliseconds; 0 min(s), 43 seconds.
```

The Utilities Module

jFuzzyMachine's Utilities Module consists of the 'Postprocessing' and the 'Add-ons' sub-modules. The postprocessing module consists of the 'Graph', 'Evaluation' and 'Dynamic Simulations' Units.

The Graph Unit

Outputs from the graph unit execution are placed in the runJFuzzUtils subdirectory. With regards to this demonstration, this would be the ./etc/projects/demo/outputs/runJFuzzy/runJFuzzUtils directory. These tab-delimited result files include:

- _demo_runJFuzzUtils.adj
- _demo_runJFuzzUtils.edg
- _demo_runJFuzzUtils.edg2
- _demo_runJFuzzUtils.fit
- _demo_runJFuzzUtils.fit2
- _demo_runJFuzzUtils.fre

The _demo_runJFuzzUtils.adj file is a directed graph adjacency matrix describing the connections in the inferred network. A connection between two nodes is indicated by 1 and 0 vice versa. Features in the rows are the inputs while those in columns are the output nodes. The _demo_runJFuzzUtils.edg and _demo_runJFuzzUtils.edg2 result files are about the same. Describing the edges in the inferred network, they both have the columns; From, To, Rule, and Weight in common. These correspond to the input node, output node, fuzzy logic rule associating the input with the output node, and estimated model fit respectively. The 'HashCode' column in the ".edg" file is only included for programmatic debugging. Likewise, the _demo_runJFuzzUtils.fit and _demo_runJFuzzUtils.fit2 result files are about the same. While the .fit2 reports all models above the specified fitCutOff in the jfuzzymachine.graph.config file, the .fit file reports only the best fitted model to each output node. Sampled outputs from the related demo run are shown below:

```
# _demo_runJFuzzUtils.fit
Output NumberOfFittedModels InputNodes(BestFit) Rules Fit
C2orf78 7 [SERPINB7, CYS1, MGAM] [[3, 3, 1], [1, 2, 3], [3, 1, 1]]
    0.7246192458666514
```

```
LINC00634 2 [KRT86, LINC00886, GCNT4] [[3, 1, 1], [2, 2, 3], [3, 1, 1]]
    0.6009588086189968
    ...
```

```
# _demo_runJFuzzUtils.fit2
Output InputNodes Rules Fits
C2orf78 [SERPINB7, CYS1, MGAM] [[3, 3, 1], [1, 2, 3], [3, 1, 1]]
    0.7246192458666514
C2orf78 [SERPINB7, CYS1, MGAM] [[3, 2, 1], [1, 2, 3], [3, 1, 1]]
    0.646429505593284
    ...
SERPINB7 [ROBO4, C2orf78, GCNT4] [[3, 2, 3], [3, 2, 1], [1, 3, 3]]
    0.6002583965182628
SERPINB7 [ROBO4, C2orf78, LINC00634] [[1, 2, 3], [3, 1, 1], [2, 3, 1]]
    0.6002285026737246
SERPINB7 [PTHLH, CYS1, C2orf78] [[1, 3, 1], [3, 1, 1], [3, 2, 1]]
    0.600143816707301
PTHLH [LINC00707, RUBCNL, LINC00634] [[1, 1, 1], [1, 3, 2], [2, 1, 1]]
    0.7444440833291
PTHLH [LINC00707, RUBCNL, LINC00634] [[2, 1, 1], [1, 3, 2], [2, 1, 1]]
    0.7321145800582836
    ...
```

The _demo_runJFuzzUtils.fre reports the frequency of the fuzzy rules evaluated in the inferred models with an estimated fit value above the fitCutOff. Please see Gormley et al [182] and Sokhansanj et al [380] for a detailed explanation of the rules.

The Evaluation Unit

The Evaluation Unit compares expression profile predictions by inferred models to an experiment values – either the fuzzy logic models' model-elucidating data or an independendent dataset. Its output are reported in the _demo_runJFuzzUtils.val file – an expression matrix

of predicted values of output nodes, given the values in the exprsToValidate file and the set of fuzzy logic models in the fitFile specified in the jfuzzymachine.evaluator.config.

The Dynamic Simulation Unit

The 'Dynamic Simulation Unit' implements and executes model dynamic simulations as also described in Gormley et al. The unit implements an iterative scheme to determine the state of the network at equilibrium. Simulation values are reported in the runJFuzzyUtils/simulations/ subsub-directory in the jFuzzyMachine main output directory. These are captured in the .dta and .sim files. The .dta files report the error values following each iteration, while the .sim file reports the estimate for each output node in the network at each iteration. The numerical value in the naming convention of the derived files show the column index, in the expression matrix, of the 'sample', 'perturbation', or 'time-point' from which initial values for the respective simulations were derived.

A.3.6 Add-ons

jFuzzyMachine and its outputs are designed to either be standalone resources, or be easy to integrate with other analyses pipelines and platforms. Add-ons or plug-ins provide an avenue to easily integrate additional functionalities or integrate other tools to the base application. For a better appreciation of results, we have included an example plug-in to enable some visualization of results demonstrated in this manual. As previously stated (please see publication), plug-ins can be platform dependent and may rely on secondary applications for full functionality. The plug-in bundled with jFuzzyMachine requires a UNIX-based platform or OS with the R statistical programming environment [269] pre-installed. In addition to having the R program pre-installed, the following R/Bioconductor [262] packages are required:

- optparse
- org.Hs.eg.db

- xtable
- igraph
- graph
- Rgraphviz
- pheatmap
- ReactomePA

To execute, simply run the following commands from within the jFuzzyMachine application working directory:

```
plugins/viz/rJFuzzyMachineUtilsExec.sh

plugins/viz/rJFuzzyMachineUtilsNetworkExec.sh
```

Example output figures, saved in the ./etc/projects/demo/outputs/plugins/viz/figs directory are presented below:

A.3.7 Benchmarking – Comparing jFuzzyMachine's inferred network to ARACNe's

To benchmark jFuzzyMachine and the fuzzy logic algorithm, we compared regulatory network inferred by jFuzzyMachine with that inferred by the ARACNe (an Algorithm for the Reconstruction of Gene Regulatory Networks algorithm [381], mutual information matrix of sampled features expression profile was inferred, using the R/bioconductor minet package build.mim routine and specifying the spearman option as the estimator. The minet package ARACNe algorithm implementation was used to derived weighted adjacency matrix of the inferred network. The identified edges were compared to those inferred from the best fitted models from a jFuzzyMachine inference, given the same expression profile.

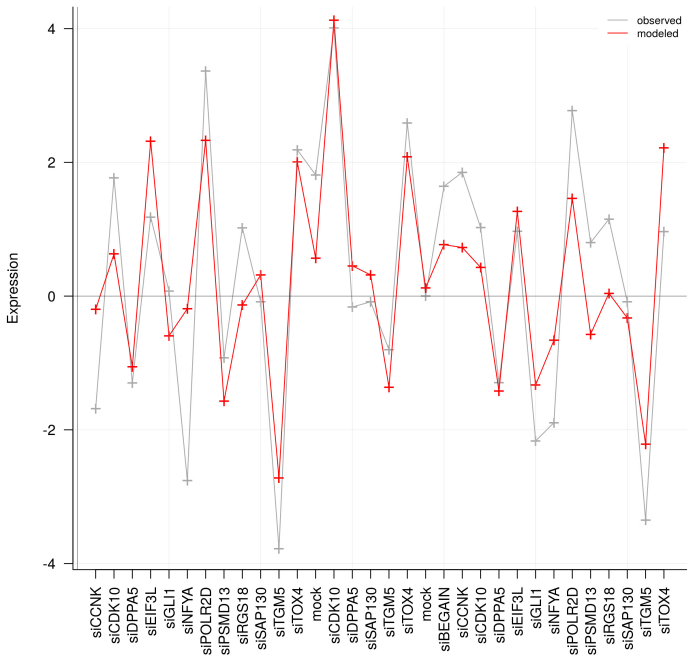


Figure A.2: Model Evaluation Plot Example. A visual evaluation of predictions of a fitted model for a sample output node, the C2orf78 gene. The estimated fit was 0.72. The input (regulatory) nodes were the genes SERPINB7, CYS1, and MGAM. The y-axis indicates the normalized expression values and the a-axis indicates the sample perturbations or treatment. Samples were reverse transfected vorinostat-resistant colon cancer, HCT-116, cell lines. Each sample was treated with the indicated small interfering ribonucleic acid (siRNA) to knockdown the respectively indicated gene products. The grey plot line shows the observed expression profile of the gene C2orf78, while the "red" line shows the predicted expression value from the expression of the regulators in the given data, and the rules associating the regulators to the output. The inferred patterns of regulation (rules) are indicated in Figure A.4. It can apparently be appreciated that the fuzzy logic model is able to tease out trend in the dataset

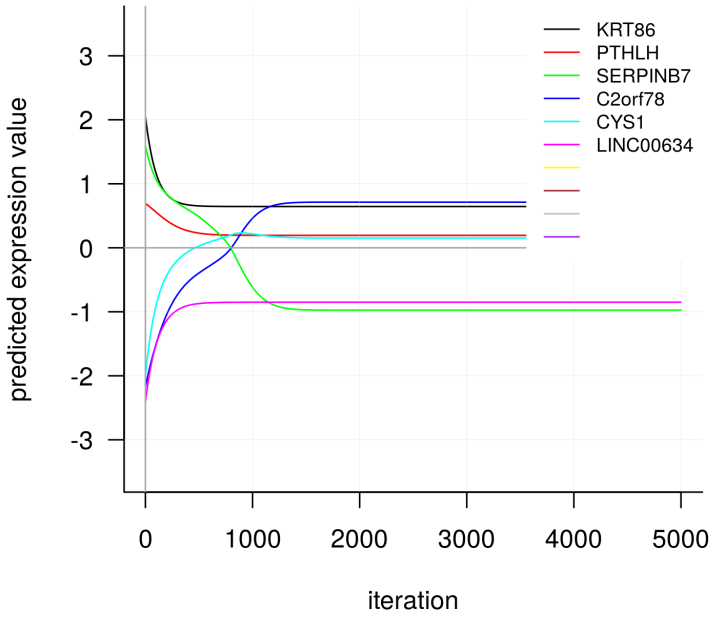


Figure A.3: A Dynamic Simulation Plot. After randomly choosing a sample from the normalized expression matrix to provide initial values of expression, and given the best fitted models, the plot shows predicted expression values for the inferred outputs KRT86, PTHLH, SERPINB7, C2orf78, CYS1 and LINC00634 over 5000 iterations. It is appreciable that the inferred network achieves an equilibrium state at a little over 1000 iterations, when a change in predicted values tend to zero

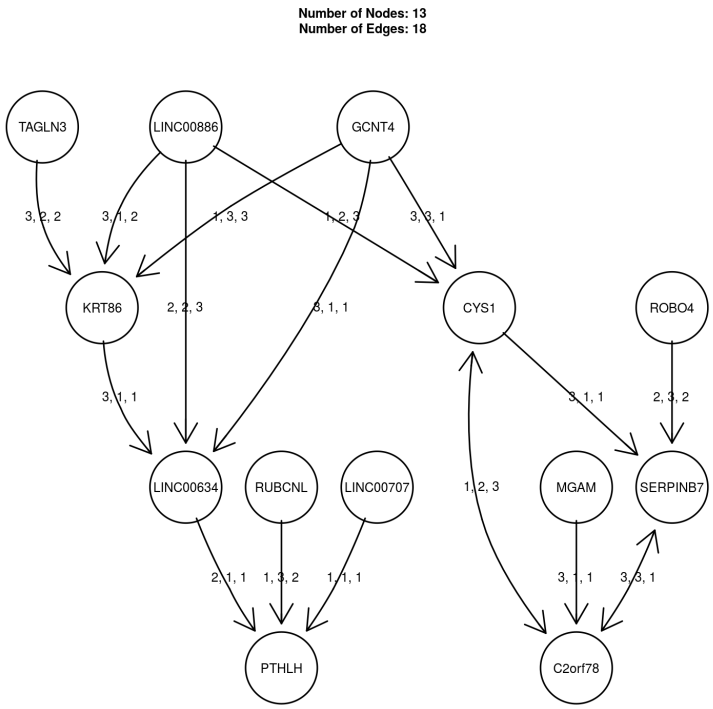


Figure A.4: The Fuzzy Logic-based Regulatory Network Inferred. A composite regulatory network is inferred from the best fitted models for each node. The inferred network consists of 13 nodes (genes), and 18 edges (regulatory connections). The arrow heads indicate the regulatory direction from the input node to the output node. The edge labels, shown by the fuzzy rules, indicate the regulatory interaction. From Gormley et al, Rule configuration is the specification of if-then relationships between variables in fuzzy space. For example, an inhibitory relationship is represented by the rule vector $r = [r_1, r_2, r_3] = [3, 2, 1]$ (i.e., if input is low (r_1), then output is high (3); if input is medium (r_2), then output is medium (2), and if input is high (r_3), then ouput is low (1). From the composite regulatory network, the regulatory effect of the MGAM gene on the C2orf78 gene is indicated by the rule $3, 1, 1$. This implies that when MGAM is low (r_1), C2orf78 is high (3); when it is medium (r_2), C2orf78 is low (1); and when MGM is high (r_3), C2orf78 is low (1). Notice that the bi-directional relationship between the pair of genes C2orf78-CYS1, and C2orf78–SERPINB7

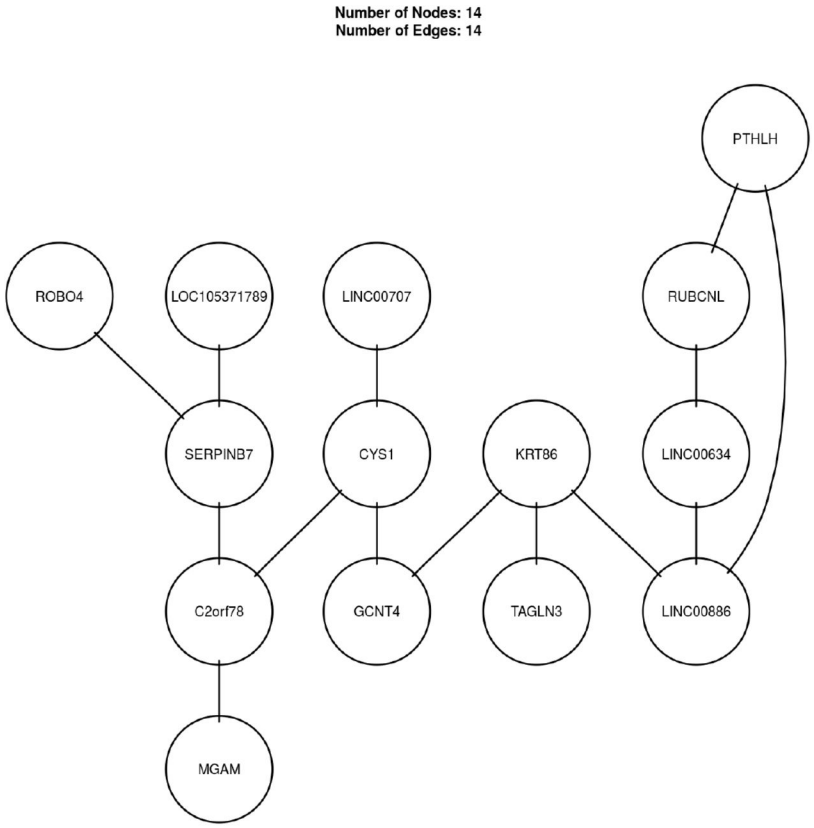


Figure A.5: The ARACNe-inferred Regulatory Network

Figure A.5 shows the inferred networks of the ARACNe algorithm. It is observed that jFuzzyMachine and ARACNe both appear to have a large overlap in the number of predicted edges (Fig. A.6). As opposed to many other network inferring algorithms, jFuzzyMachine always predict the direction of relationship (i.e. what node is regulating the other node).

A.4 Conclusion and Recommendation

The Fuzzy logic inference approach to elucidating regulatory networks, although relatively mature, has little to no readily available tool to democratize its adoption on a larger scale. The jFuzzyMachine tool fills the need for a freely available fuzzy logic-based inference system, particularly to the scientific community. It addresses in part, the apparent lack of readily available community tools, removing a limitation to the applicability and benefits of the fuzzy inference system to elucidating biological data.

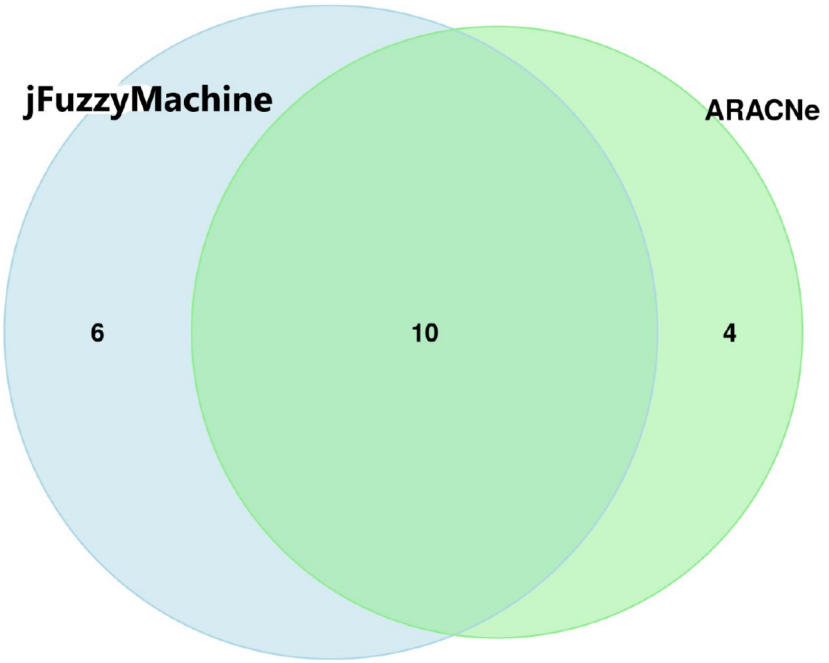


Figure A.6: ARACNe vs jFuzzyMachine identified network edges

A.5 Future Direction

Current implementation of the jFuzzyMachine implements a few of the available Fuzzy logic-based inference methods applicable to biological data in the scientific literature. In tandem with in-house development efforts and biological validation of advances to the Fuzzy logic-based methods, we anticipate to continually include added functionalities. With our modular design and plan to accommodate third-party add-ons, we hope to facilitate community contributions and a scientific ecosystem of adopters.

Appendix B:
Time Complexity of the Fuzzy Logic Inference Algorithm

B.1 Introduction

Although it circumvents hyperparameter estimation of ordinary differential equation (ODE) models and the potential problems that are associated with inaccurate parameter estimates, the computational complexity of a fuzzy logic regulatory model, particularly at higher order of interactions, quickly approaches these more complicated models. Approaching computationally intractable problems, the benefits inherent in the simplicity and strength of fuzzy logic models become undermined. To facilitate higher order model inferences in significantly faster computational time, we performed a computational time complexity analysis of a classical fuzzy logic regulatory model inference system – one that implements the union rule configuration and, we developed and implemented a "multistaged hyperparallel" optimization approach. For a sampled inference problem, the "multistaged hyperparallel" optimization approach is demonstrated to significantly shorten time to model inference from about 485.6 hours (20 days) to approximately 9.6 hours (0.4 days).

B.2 Method

The classical fuzzy logic triplet model of Woolf and Wang is reported to run on the order $O(n^3)$. Where n is the number of interacting molecules – a very conservative estimate. It accounts for only the number of fuzzy rule evaluations performed for a specific combination (activator-repressor-target), of a particular set of triplet. It does not account for those of other combinations nor does it account for all other possible triplets. These others, considered together, can have a combinatorial explosion-like growth function that may quickly become significant in comparison to that observed with the rules evaluated with increasing n. Employing the union-rule configuration (URC), Sokhansanj et al were able to reduce

the complexity of Woolf and Wang's solution from $O(m^{NN})$ to $O(m^N)$. Where N is the number of (input) genes regulating an output gene and m is the number of possible rules describing the effect of each single input gene on an output gene. The number of possible rules for each gene-gene interaction (m) is given by n^n, where n is the number of fuzzy sets that describe the state of a variable[49]. Similarly, this is a very conservative estimate. It accounts for only the number of fuzzy rule evaluations performed for a specific combination of a particular set of inputs (regulators) and output genes.

B.3 Theoretical Analyses

To analyze the computational time complexity of the Sokhansanj approach, a pseudocode is presented here

The Exhaustive search

1. Read-in the configuration file
2. Initialize object
3. Initialize table of expression
4. Initialize fuzzy Matrix (fuzzified values of expression values)
5. // for a constant time access to fuzzy sets of expression values
6. // do exhaustive search:
7. get the output nodes (output genes), $ON\_1, ON\_2, ON\_3 \cdots ON\_N$
8. // these may be all the genes in expression matrix or a pre-specified number
9. for each of the output gene node:
10. get other genes (potential inputs to the current output node)
11. get the 'number of input' nodes to consider

12. // may consider a maximum number of input nodes IN_Max

13. // defaults to a specific user specified number of inputs

14. // do deeper search:

15. get the desired e-value cutOff (e-cutoff)

16. get combinations (permutations) of input nodes; $CIN\_1 \cdots CIN\_p$

17. get output gene expression values

18. get mean expression value of output gene

19. get the sum of squared deviations (dss) of output gene expression values

20. // get combinations of inputs

21. for each combination (of inputs):

22. get all possible combinations of fuzzy rule to evaluate //nested for-loops

23. for each possible combinations of fuzzy rule

24. instantiate a string array for the input genes

25. // compute residuals

26. for each expression value of the output gene across all samples, time series or perturbations

27. get input genes

28. get the fuzzy set values for respective input genes

29. perform a union rule configuration (URC) evaluation

30. defuzzify aggregate fuzzy set

31. compute residual squared sum (rss)

32. // sum squared residual

33. compute fit (error) $= 1 - (\frac{\text{rss}}{\text{dss}})$

34. if computed fit is greater than or equals e-cutoff

35. populate fuzzy 'rule' arrays with valid rule instances...

36. instantiate a result object

37. // may add result object into a collection of result objects

38. print acceptable result

39. end_if

40. end_for

41. end_for

42. end_for

43. end_for

From a set of output nodes (gene features to be included in the derived regulatory network), the algorithm independently and exhaustively search for models (combinations of inputs to output), across samples, that meet prespecified fit cut-off (lines $6 - 45$). From a calculation of operations in the outlined pseudocode, time complexity is approximately:

$$N \cdot m^n \cdot \binom{N}{n}$$

Where

N is the number of output genes being considered.

m is the number of possible rules for each gene-gene interaction, this is the square of the

Table B.1: A Fuzzy-logic theoretical time complexity estimates

n	m^n	$\binom{N}{n}$	$m^n \cdot \binom{N}{n}$	$log_2(m^n \cdot \binom{N}{n})$
1	27	50	1,350	4.75
2	729	1,225	893,025	9.5
3	19,683	19,600	385,786,800	14.26
4	532,441	230,300	122,390,862,300	19.02
5	14,348,907	2,118,760	30,401,890,195,320	23.77

number of fuzzy sets that describe a variable. For a three fuzzy sets (LOW, MEDIUM, and HIGH) model, this would be 3^3, which is 27. And,
n is the number of input nodes being considered.

Given the following number of inputs, and fifty output nodes, analytical estimates of computational time complexity is estimated in the table below:

Table B.1 shows the analytical estimates of computational time complexity in milliseconds. Note that the total number of outputs, N, being constant, was omitted in estimating big O, $N \cdot m^n \cdot \binom{N}{n}$.

Figure B.1 shows a plot of the logarithm of the analytical estimate of time complexity (calculated cost) as a function of inputs to the algorithm. The derived log estimates were fitted using a simple linear regression model to obtain a slope, estimated to be 8.2146. This implies that for every additional input considered, the computational time complexity grows by approximately eight folds, if every other variable or factor remains constant.

B.4 Empirical Analyses

To investigate how well our estimates capture real world situations, we ran our implementation of the algorithm with up to four inputs. Table B.2 shows the execution time, considering only one output node. Aside from the differing number of input nodes considered, all other factors were kept the same. The computation experiment was performed

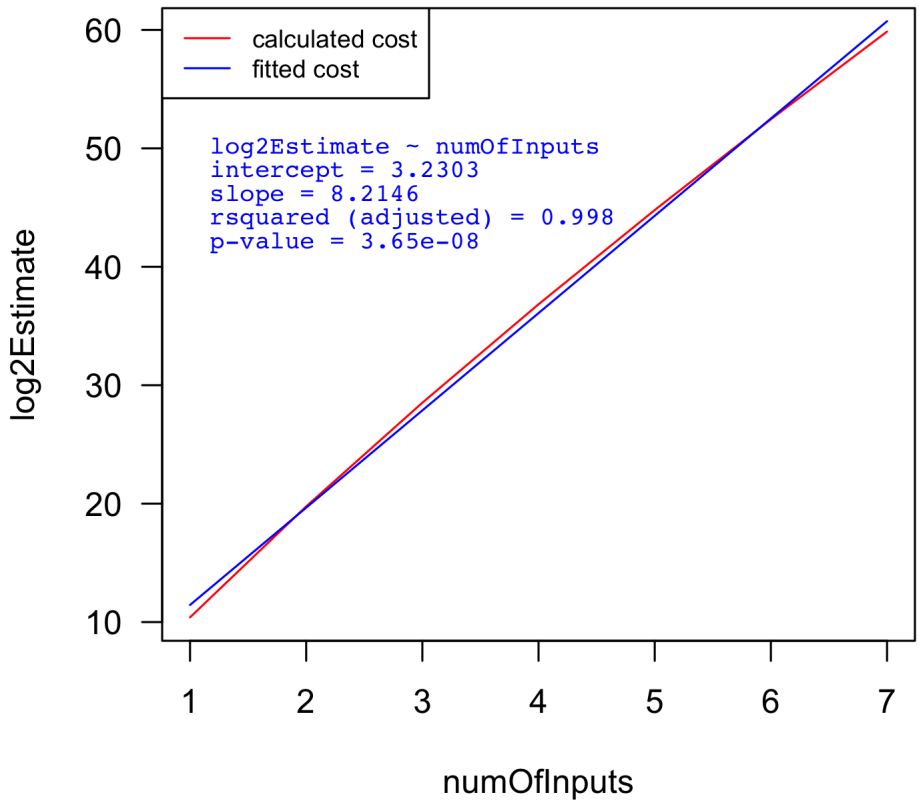


Figure B.1: Logarithm of the analytical estimate of time complexity versus number of inputs

Table B.2: Empirically derived execution time

n	Execution time (in milliseconds)	log_2(Execution time)
1	169	7.40
2	1,011	9.98
3	127,132	16.96
4	43,722,655	25.38

on a compute node of the Frederick National Laboratory High Performance Computing Environment with 32 cores and 18GB of available runtime memory. A compute node is an x86_64 Genuine Intel ®Xeon ®Gold 6150 CPU @2.70GHz. – See section on optimization (B.5.1, The Multi-staged, Hyper-parallel Optimization)

We also fitted the observed log value of execution time using a simple linear regression model to obtain a slope (growth rate) (Figure B.2). Though empirical growth rate appears to be less than that estimated from a complexity analyses of the algorithm, the nature of the curve beyond three inputs appear to tend towards the analytical estimates (big O, the asymptotic upper bound on the function). More importantly, the growth rate beyond two inputs appears almost parallel to that of the analytical estimates (Figure B.3). As previously mentioned, it is also observed that a m^n computation time complexity specification underestimates the real world nature of the algorithm (Figure B.4).

B.5 Improving Time Complexity

Alluded to in Woolf and Wang's, the fuzzy logic algorithm consists of solving a large number of smaller, independent comparisons, and it lends itself to parallel computing. This is presumed to potentially scale nearly linearly with the number of available processors. In recent times, more readily available workstations with much faster clock-speeds and multicore/-multithreaded abilities, including ready access to high performance computing environments present opportunities to investigate higher degrees of interactions at individual nodes of a regulatory network with a fuzzy logic model.

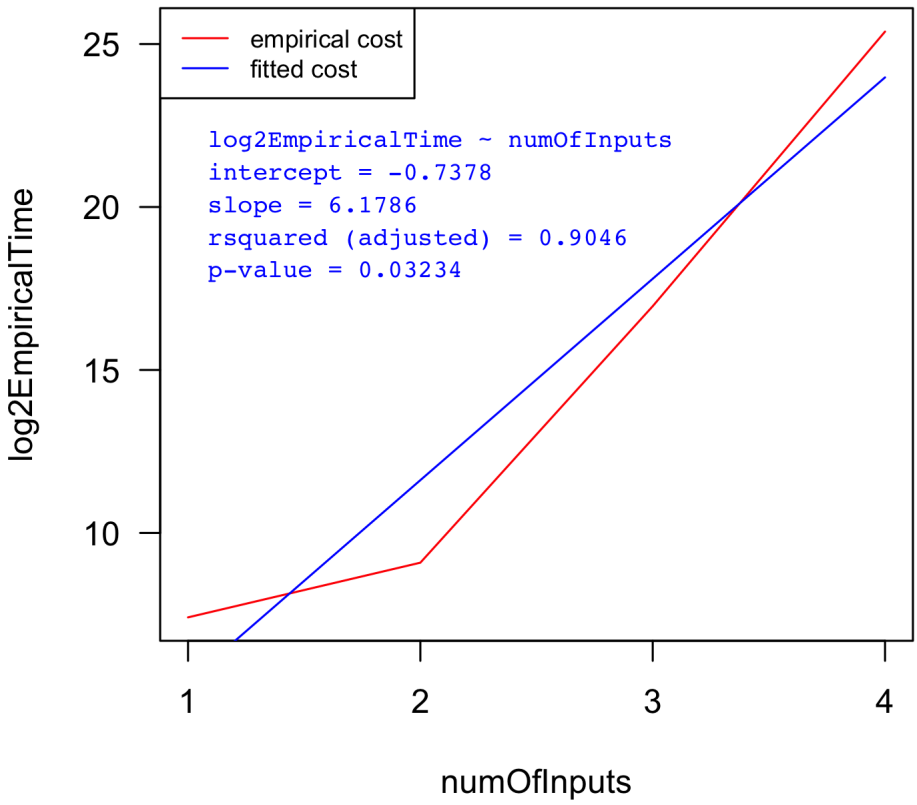


Figure B.2: Logarithm of execution time versus number of inputs I

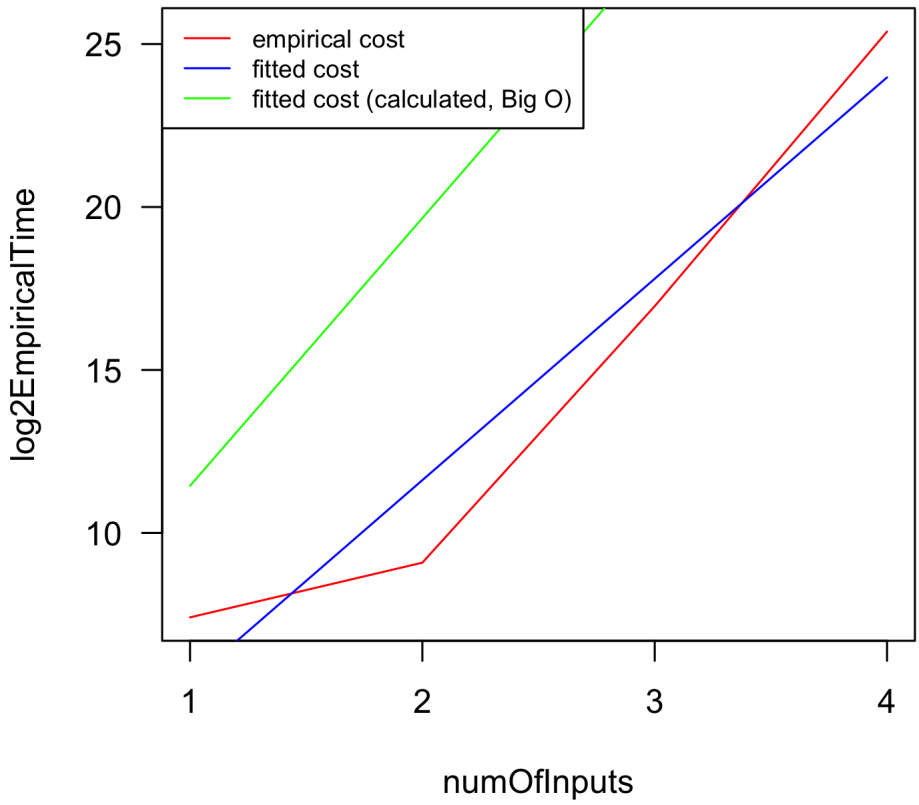


Figure B.3: Logarithm of execution time versus number of inputs II

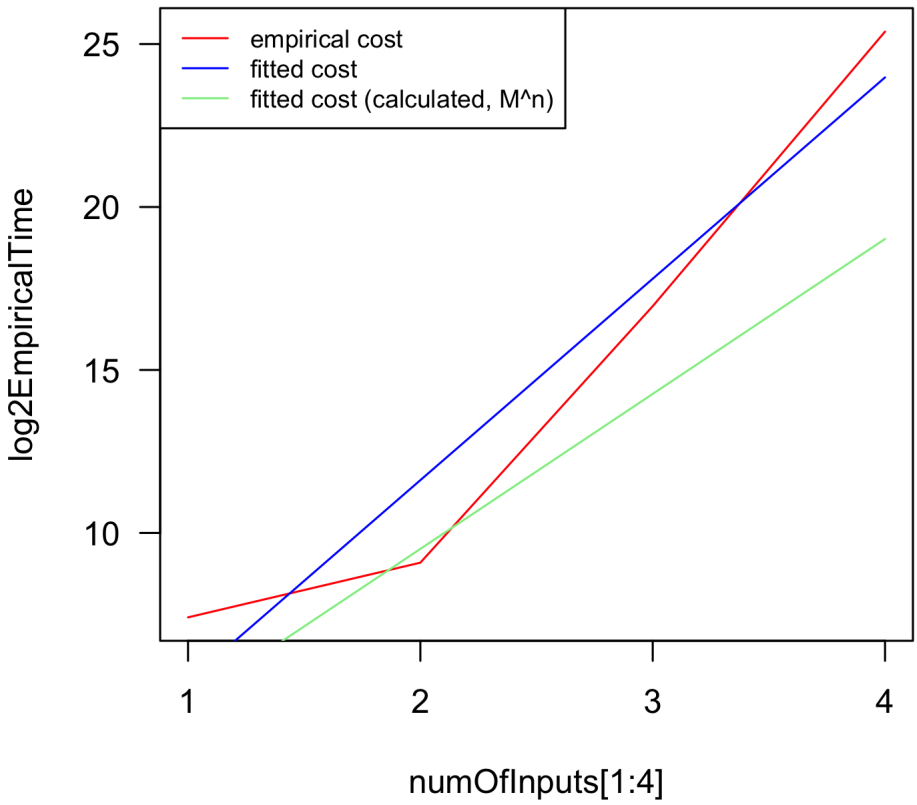


Figure B.4: Logarithm of execution time versus number of inputs III

B.5.1 The Multi-staged, Hyper-parallel Optimization

The Multi-staged, hyper-parallel optimization approach presented here is analogous to the "divide-and-conquer" computation algorithm-design paradigm. A "divide-and-conquer" algorithm-design paradigm entails breaking down a complex problem into smaller and easier entities. It involves dividing the complex problem into as many subproblems as is simple enough to solve and, combining the solutions to the subproblems to get a solution to the original problem. The divide-and-conquer paradigm has been utilized in many serial and serial-like algorithms, particularly the recursion based mergesort, binary search, quicksort and many others, including efficient algorithms for computing the discrete Fourier transform (FFT) [382, 383]. With respect to parallel algorithms, the MapReduce programming model [384][385][386] can be considered a "divide-and-conquer" algorithm-design paradigm.

Multi-Staging

With respect to modelling multithreaded (and by extension, parallel) executions, Cormen et al suggested that it helps to think of a multithreaded computation as a directed acyclic graph $G = (V, E)$, called a computation dag [383]. Conceptually, the vertices in V are instructions (and data objects), and the edges in E represent dependencies between instructions (and data objects), where $(u, v) \in E$ means that the set of instruction u must execute before instruction v [383]. A closer examination of our outlined fuzzy logic with union rule configuration algorithm pseudocode shows multiple lines of dependent instructions and blocks of potentially parallel operations (Figures B.5 and B.6). To facilitate a hyper-parallel processing in a high performance computing (HPC) environment, we staged the runtime SLURM batch schedule [387, 388], to achieve a distribution across many more processor cores simultaneously. The SLURM scheduler, a de facto manager on many HPC environments, facilitates dynamic multithreading (parallel processing), allowing computation to specify parallelism without worrying about communication protocols between environment nodes, load balancing, and other peculiarities of static-threads.

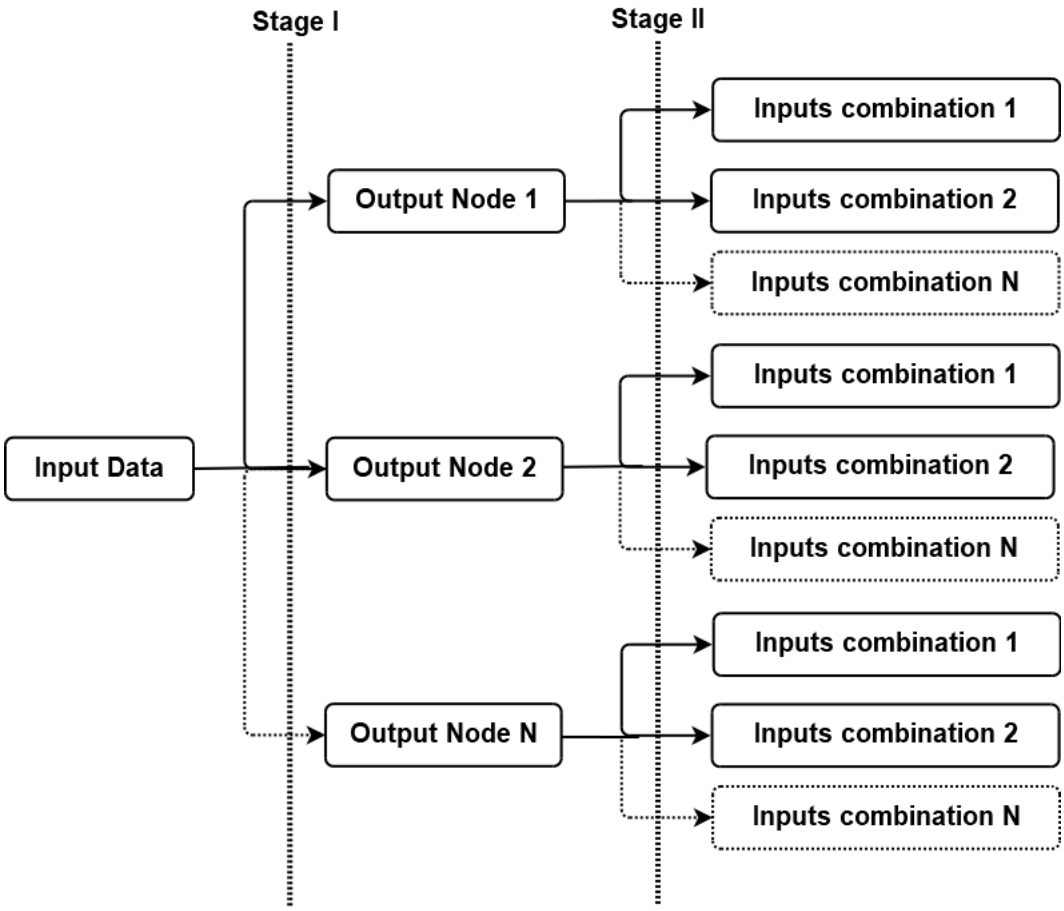


Figure B.5: Computation dag (directed acyclic graph) I

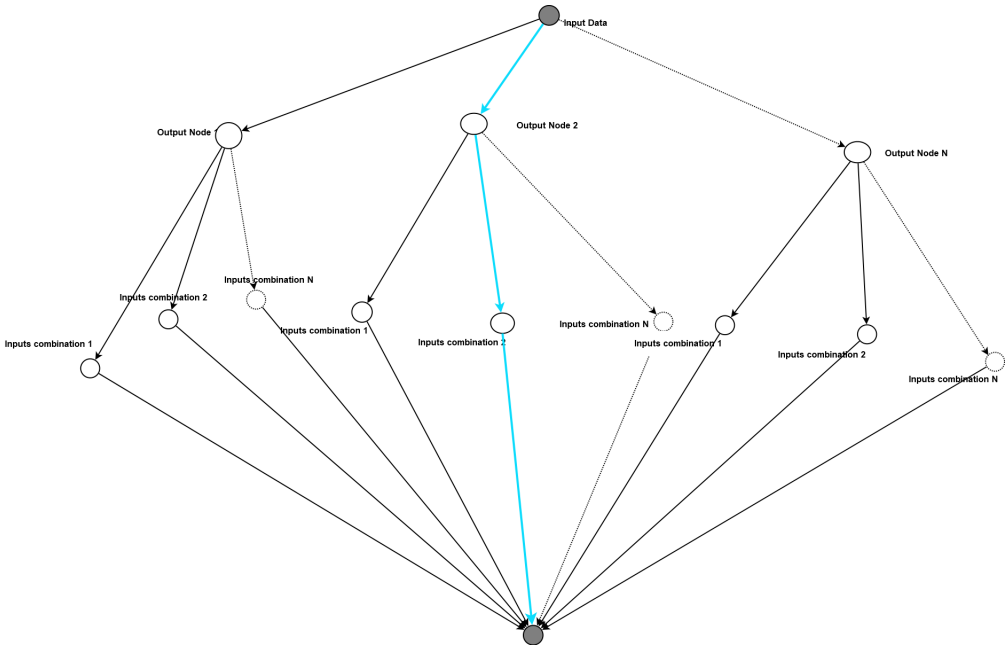


Figure B.6: Computation dag (directed acyclic graph) II

Logically, if a vertex (set of instructions and objects, v) has a direct path from another vertex (set of instructions and objects, u), both processes, u and v are described as (logically) in series. But, (logically) in parallel if not (Cormen et al. 2009). Thus, it appears reasonable to stage runtime SLURM batch schedule (at indicated staged points, Stage I and II, in Figure B.5) because downstream (child) processes in the computation dag (Figure B.6), are independent of one another (logically in parallel), and only dependent on antecedent (parent) processes (set of instructions and or objects).

Measuring performance improvement (theoretical efficiency)

The discourse on the scalability of a parallel or multithreaded computation work and the best model to evaluate its performance still persists. This dates back to about five or six decades. Sometimes referred to as recurring, re-stirred by worries about the pessimistic implications of Amdahl's law [389]. Gene Amdahl on what became known as Amdahl's law made a submission for a single processor approach to large scale computing, arguing that

for most applications, there exist a sequential potion that cannot be parallelized. He argued that, with an increasing number of processors, this sequential portion may constitute up to 50%–80% of the total execution time, and thus have a diminishing effect [389,390]. Amdahl's law is also referred to as the fixed-size speedup model [391–394]. It implies that, if a portion of a computation, f, can be improved by a factor m, and there exists another portion that cannot be improved, then the portion that cannot be improved will quickly dominate the performance, and further improvement of the improvable portion will have little effect [389].

$Speedup_{Amdahl} = \frac{1}{(1-f)+\frac{f}{m}}$

Where f is a parallelizable portion, and m the number of processors. Note that as $\lim_{m\to\infty}$, $Speedup_{Amdahl} = \frac{1}{1-f}$

Together with colleagues at the Sandia National Laboratories working on a 1024 processor system, Gustafson et al demonstrated that the assumptions of Amdahl's argument were inappropriate to describe observed results with massive parallelism [395–397]. Identifying the shortfall in an implicit assumption in Amdahl's law - that the number of processors is independent of size of the problem, Gustafson proposed that it would be more realistic to assume run time, not problem size is constant - the fixed-time speedup model [395]. Fixed-time speedup, Speedup_{FT} is given as:

$$\text{Speedup}_{FT} = \frac{\text{Sequential Time of Solving Scaled Workload}}{\text{Parallel Time of Solving Scaled Workload}}$$

If an original workload, ω, and a scaled workload $\omega\prime$, finish the same amount of time with sequential processing and parallel processing with m processors respective; with an assumption that scaling is in the parallel part only, it implies

$\omega\prime = (1-f)\omega + fm\omega$

Therefore,

$$\text{Speedup}_{FT} = \frac{\text{Sequential Time of Solving } \omega\prime}{\text{Parallel Time of Solving } \omega\prime}$$

$$\text{Speedup}_{FT} = \frac{\text{Sequential Time of Solving } \omega\prime}{\text{Sequential Time of Solving } \omega}$$

$$= \frac{\omega\prime}{\omega}$$

$$= \frac{(1-f)\omega + fm\omega}{\omega}$$

$$= (1-f) + mf$$

Known as the Gustafson's law, the above equation's implication can be stated as, "the fixed-time speedup is a linear function of m if the workload is scaled to maintain a fixed execution time.

On the assumption that many applications are unable to scale up to meet the time bound constraint due to some physical constraints, Sun and Ni proposed the memory-bounded speedup model. Summarized as Sun and Ni's law. Given that $y = g(x)$ is the parallel workload increase factor as the memory capacity increases m times; $\omega = g(M)$, and M is the memory capacity of one node.

$$\text{Speedup}_{MB} = \frac{(1-f)\omega + f \cdot \bar{g}(m)\omega}{(1-f)\omega + \frac{f \cdot \bar{g}(m)\omega}{m}}$$

$$= \frac{(1-f) + f \cdot \bar{g}(m)}{(1-f) + \frac{f \cdot \bar{g}(m)}{m}}$$

Where $\bar{g}(m)$ is the power function with a coefficient of 1. This generalizes Amdahl's and Gustafson's laws, which are both special cases where $\bar{g}(m) = 1$ and $\bar{g}(m) = m$ in respective cases. Sun and Ni's model gives a higher speedup than both pure Amdahl's and Gustafson's speedup model.

For a theoretical analysis of our fuzzy inference engine for a regulatory network multistage, hyperparallel algorithm, two metrics – "work" and "span" are useful, borrowing from Cormen et al [383]. Work is defined in this case as the total time to execute the entire computation on one processor. For our computation dag (Figure B.6) in which each edge is assumed to take a unit time, work is equivalent to the total number of vertices. The span is the longest time to execute the strands (a chain of instructions containing no parallel control) along any path in the dag. For our dag, the span equals the number of vertices on a longest or critical path in the dag (colored path in Figure B.6). For our example computation dag, the total number of vertices would be given as $Outputs \cdot (\binom{N-1}{n} + 1)$, where $Outputs$ is the number of output genes being considered, N is the number of genes in the network, and n is the number of input genes being considered. For a single output and two input genes from a fifty genes set, our computation dag would have approximately 1177 vertices (work of 1177 time units) and a span of 2 vertices (2 time units). Figure B.7 shows a plot of the theoretical or expected speedup that can be achieved at varying number of available computing cores in a high performance computing (HPC) environment, not considering node, scheduling, I/O, memory or other possible computational overheads. Figure shows theoretical speedup for both Amdahl and Gustafson's models.

Speedup observed using Gustafson's model does appear to increase linearly with available computing cores. However, it appears to approach an asymptotic maximum with Amdahl's model. Cormen et al described "parallelism" of a multithreaded (and by extension a parallel) computation. This they described as the average amount of work that can be performed in parallel for each step along the critical path. Its estimation is described as the maximum (upper bound) speedup that can be achieved on any number of processors

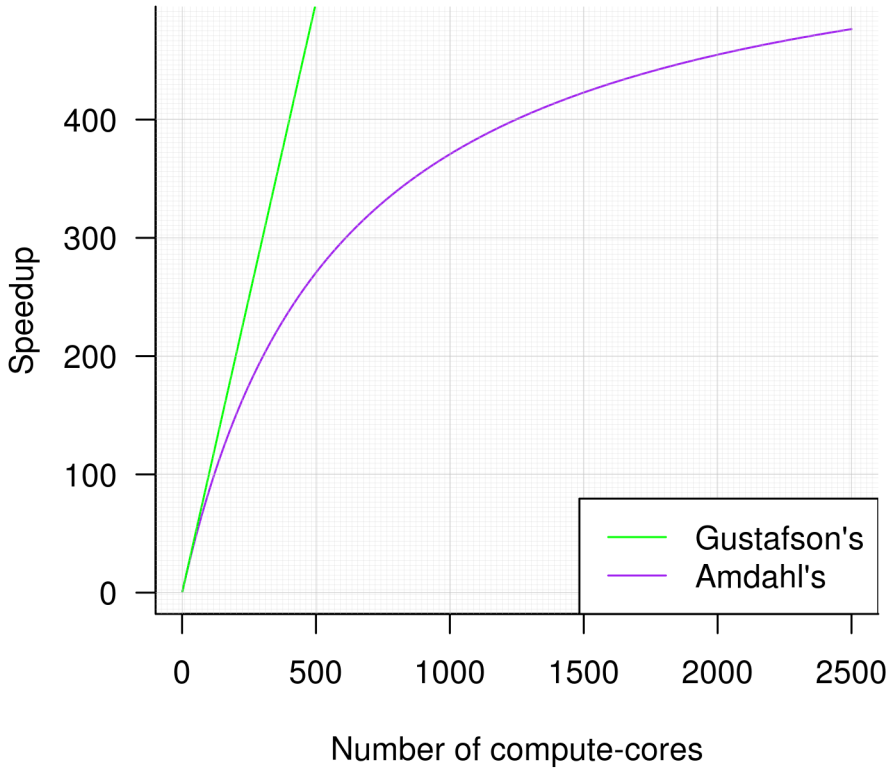


Figure B.7: Parallel speedup as a function of computing cores

Table B.3: Empirical execution time of improved algorithm

Number of cores	Execution time (in milliseconds)
1	340355
2	126925
3	95324
4	79626
5	66694
6	60592
7	52900
8	49675
9	44976
10	40064
11	37628

[383]. Given that our work estimate from our computation dag (Figure B.6) is $T_1 = 1177$ and span (irrespective of the number of available processors) is $T_\infty = 2$. Cormen et al defined parallelism as $T_1/T_\infty \approx 589$ may well approximate the possible speedup upper bound using Amdahl's model.

Measuring performance improvement (empirical efficiency)

To evaluate how well our theoretical evaluation mirrors a real world situation. We ran our algorithm with the same parameters on a multicore machine 32bit/64bit x86_64, GenuineIntel ®CPU@1.80GHz. The parameters were – a single output gene, two input regulatory genes and a fifty genes set. We observed the execution time using 1 to 11 computing cores, and calculated the speedup of a computation by the ratio $T1/T_P$, where T1 is the algorithm's execution time with just one core and T_P is the execution on a specified number of processors. Table B.3 shows observed execution time in milliseconds. Figure B.8 is a barplot of the execution times. Figure B.9 shows a plot of speed up versus available computation core (processor). The fitted line of the plot shows an almost linear growth curve. The largest change in speedup gradient appears to be between one and two cores

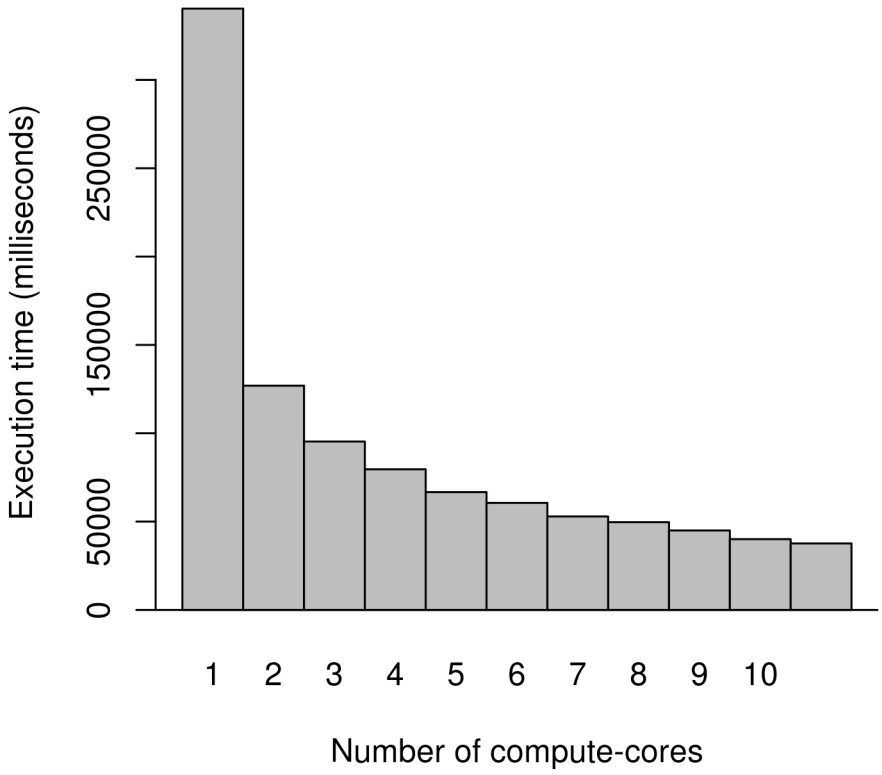


Figure B.8: Empirical execution time of improved algorithm

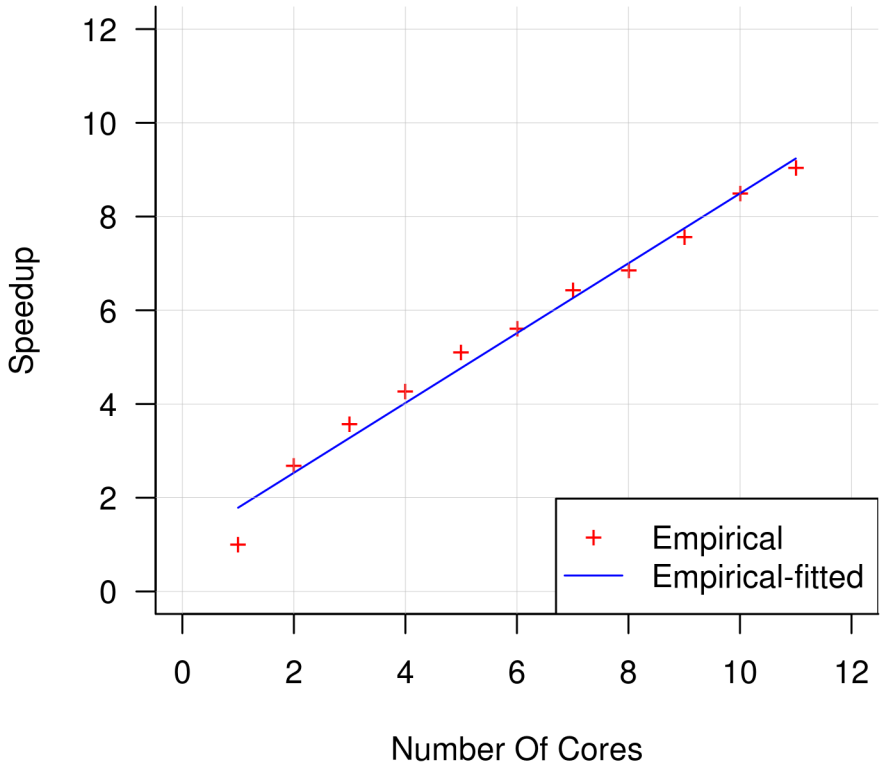


Figure B.9: Empirical speedup observed with improved algorithm

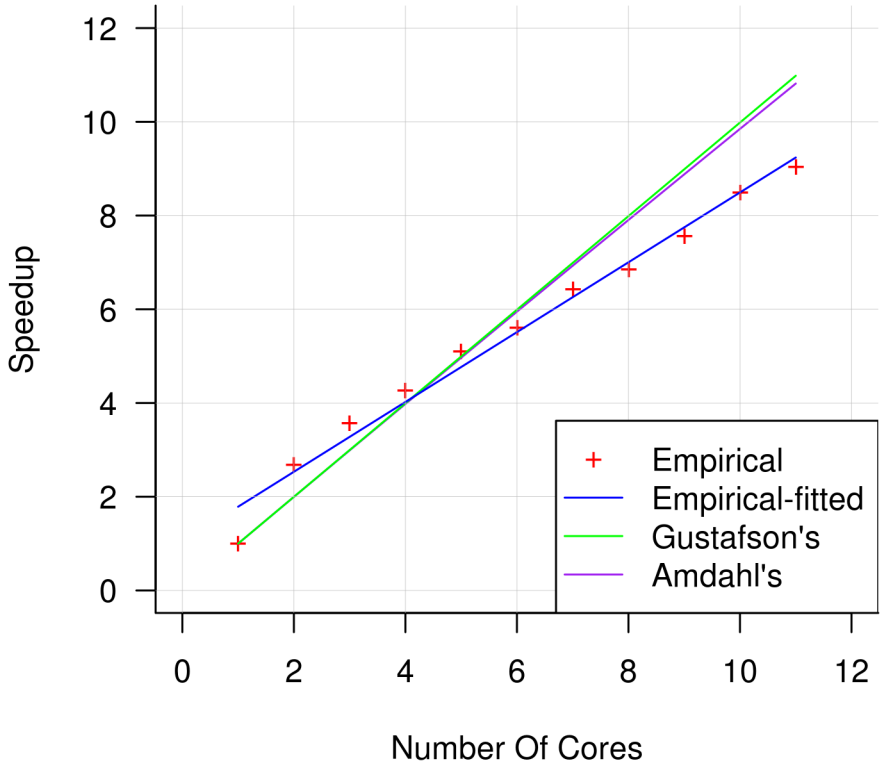


Figure B.10: Empirical speedup observed with improved algorithm compared to Amdahl's and Gustafson's approximations

Figure B.10 overlays Amdahl's and Gustafson's model speedup estimates at the respective number of cores. Empirically observed speedup at one and two processing cores appear higher that both predictions of Amdahl and Gustafson. The generally slower rate of change of the curve quickly brings the speedup gain per increase in core to below Amdahl and Gustafson's. At a lower number of processing cores, both Amdahl's and Gustafson's estimates appear to trend together. The curves however begin to diverge at about 8 or 9 processing cores.

To show the increase in efficiency obtained with our approach, we applied it to a sample Fuzzy logic based regulatory network inference problem – one with 50 separate output genes, 4 input regulatory genes from a fifty-genes set. From Figure B.11, the "multistaged hyperparallel" optimization approach is demonstrated to significantly shorten time to model inference from about 485.6 hours (20 days) to approximately 9.6 hours (0.4 days).

Representing an almost 50 fold increase in speedup, which almost correspond to the number of output genes being considered and also corresponds to the number of Stage I grouped computation units (see Figure B.5, and section on multistaging, B.5.1), the multistage hyperparallel approach tend to keep execution time constant for every increase in output genes considered to gain a corresponding fold speedup, provided all other parameters remain the same. It may well be assumed that the multistaged hyperparallel approach tend to reformulate the fuzzy logic computation and inference problem from that which obeys the Amdahl's law to one which approximates the Gustafon's model.

B.6 Conclusions

The fuzzy logic regulatory network inference method is a simple yet powerful approach to elucidating interacting molecule in regulatory networks whose efficiency becomes undermined by high computational complexity at higher order interacting molecule inference problems. The multistaged hyperparallel approach presented and our study demonstrates

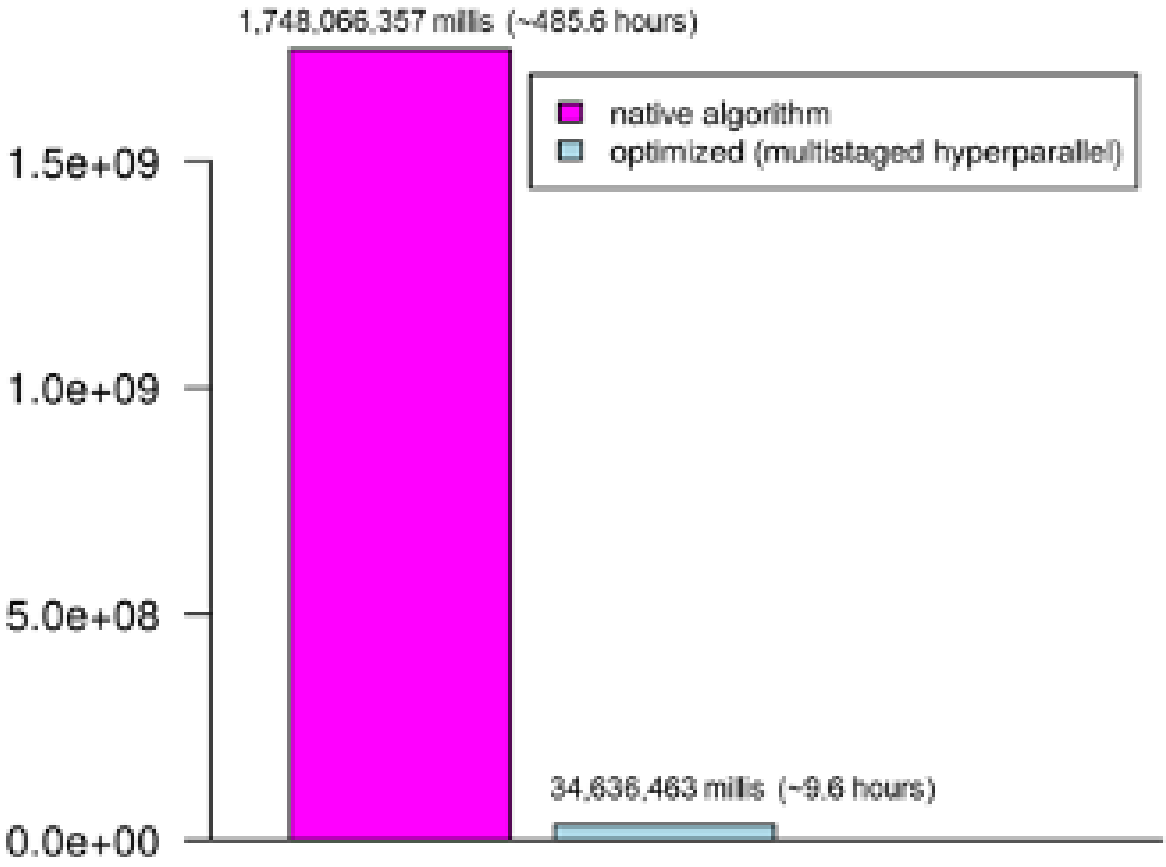


Figure B.11: Comparison of execution time (in milliseconds) between "multistaged hyperparallel" optimized algorithm and the optimized native algorithm. For a sampled inference problem, the "multistaged hyperparallel" optimization approach is demonstrated to significantly shorten time to model inference from about 485.6 hours (20 days) to approximately 9.6 hours (0.4 days)

that though the fuzzy inference system is amenable and readily scales with additional compute cores, the speedup gained per unit increase in compute core, within a high-performance computing environment diminishes and more likely to approach an asymptotic maximum, tending to more closely mimic Amdahl's model than the Gustafson's model. The multi-staged hyperparallel optimization approach presented significantly improves computation time, by reformulating, in practical terms, the inference problem from what follows the Amdahl's model to that which approximate Gustafson's.

Appendix C: *In-Silico* Validation of Synthetic Lethal Partners to Histone Deacetylases (HDACs)

C.1 Introduction

Here we attempt some validations of predicted potential synthetical lethal partners to histone deacetylases, derived from our previously inferred regulatory network. We had hypothesized that vorinostat, a histone deacetylase inhibitor resistance is a result of upregulation of embryonal cellular differentiation processes. We had employed a knowledege-guided fuzzy logic regulatory inference method to elucidate these mechanistic relationships. We validated inferred regulatory models in independent datasets. And, we evaluated the biomedical significance of key regulatory network genes in an independent clinically annotated dataset. We found no significant evidence that vorinostat resistance is due to an upregulation of embryonal gene regulatory pathways. Our observation rather support a topological rewiring of canonical oncogenic pathways around the PIK3CA, AKT1, RAS/BRAF etc. regulatory pathways. Reasoning that significant regulatory network genes are likely implicated in the clinical course of colorectal cancer, we show that the identified key regulatory network genes' expression profile are able to predict short- to medium-term survival in colorectal cancer patients – possibly providing a rationale basis for prognostication and potentially effective combination of therapeutics that target these genes along with vorinostat in the treatment of colorectal cancer. Here we posit that the significant regulatory network genes (topranked, by estimated node importance) are synthetical lethal partners to histone deacetylases.

C.2 Methods

Levaraging available large scale experiment datasets, we assessed the effect of silencing each of these genes (fuzzy logic regulatory network topranked features, Table C.1) in cancer cell lines where histone deaceytylase is mutated or silenced, employing previously and recently

Table C.1: Table of Fuzzy Logic Regulatory Network Top Features by Node Importance

SYMBOL	ENTREZID	GENENAME
UBC	7316	ubiquitin C
PTEN	5728	phosphatase and tensin homolog
SMAD2	4087	SMAD family member 2
LMO7	4008	LIM domain 7
GNAZ	2781	G protein subunit alpha z
POLR2D	5433	RNA polymerase II subunit D
TP53	7157	tumor protein p53
AKT1	207	AKT serine/threonine kinase 1
RIMBP3	85376	RIMS binding protein 3
CCNK	8812	cyclin K
TNS1	7145	tensin 1
PSMD13	5719	proteasome 26S subunit, non-ATPase 13
PXN	5829	paxillin
RIMBP3B	440804	RIMS binding protein 3B
RIMBP3C	150221	RIMS binding protein 3C
APC	324	APC regulator of WNT signaling pathway
GALNT12	79695	polypeptide N-acetylgalactosaminyltransferase 12
MAPK1	5594	mitogen-activated protein kinase 1
PARVG	64098	parvin gamma
CTNNB1	1499	catenin beta 1

described mixed and quantitative (multi-omics) approaches. Specifically, we employed the [398–400]

- SynLethDB, synthetic lethality database toward discovery of selective and sensitive anticancer drug targets
- DiscoverSL, An R package for multi-omic data driven prediction of synthetic lethality in cancers, and
- SL-BioDP, Synthetic Lethality Bio Discovery Portal (SL-BioDP)

C.2.1 SynLethDB

We downloaded and explored stored data from the SynLethDB database [398] for any reported evidence of synthetic lethality between the HDACs (histone deacetylases) and

members of the top-ranked features from our derived regulatory network. SynLethDB is a comprehensive database, containing epistatic i.e synthetic lethal pairs of genes retrieved from biochemical assays, other related databases, computational predictions and text mining results on human and four other model organisms – the mouse, fruit fly, worm and yeast. SynLethDB computes a confidence score by integrating individual scores derived from different evidence sources. We focused on reported SL pairs in humans.

SynLethDB assigns quantitative scores based on experimental methods employed to derive SL partners. For multiple pieces of evidence of the same type (e.g. experimental evidence) supporting a specific SL pair, the probability disjunction formula was used to combine the individual scores:

$$s = 1 - \prod_{i=1}^{n}(1 - p_i) \tag{C.1}$$

Where s is the integrative score corresponding to the experimental evidence, p_i is the individual score, and n is the total number of pieces of experimentally supporting evidence. For an integrated score from different modality of evidence, SynLethDB introduced weight factors and to obtain a normalized score between 0 and 1 (score closer to 1 represents higher confidence), the normalized weighted sum is estimated as:

$$S = \frac{w_m s_m + w_d s_d + w_p s_p + w_t s_t}{w_m + w_d + w_p + w_t}, \tag{C.2}$$

Where S is the integrative confidence score; w_m, w_d, w_p and w_t are the weight factors of biochemical experiment, other related databases, computational prediction and text mining-based evidence; s_m, s_d, s_p and s_t are corresponding individual scores.

C.2.2 DiscoverSL

In addition to validation using curated biological evidences, we sought validation using described computational methods for predicting synthetic lethality by Das et al, the DiscoverSL. Similar to DAISY[401,402], DiscoverSL[399] is a multi-omic data-driven approach, which uses the cancer genome atlas' (TCGA) [403] data to predict synthetic lethal interactions. DiscoverSL seeks to identify clinically relevant lethal interactions. DiscoverSL combines identified mutations, copy number alterations and gene expression data from TCGA to develop a multi-parametric random forest classifier. *In-silico* evaluation of predicted synthetic lethal genes is tested using shRNA and drug screening data from cancer cell line databases. And clinical significance of prediction is evaluated using the Kaplan-Meier analysis of clinical outcome in patients with mutation in primary gene versus over or under-expression in the synthetic lethal or interaction gene. Against the positive lethal interactions reported in the SynLethDB database [398], DiscoverSL outperforms predictions by the comparative DAISY algorithm [401, 402] – among 32 literature-reported SL interactions from a benchmark data, DiscoverSL could identify 28 SL interactions, while DAISY could identify only 11 interactions. Assuming that interacting genes will tend to be functionally associated and loss of both will be lethal to the cancer cell, for each gene pair (primary gene and interactor gene), in a specific cancer type, DiscoverSL describe four predictive features (Figure C.1):

1. DiffExp – Differential expression of interactor gene based on mutation status of the primary gene, p-value

2. Exp.correlation – Pearson's correlation co-efficient between the expression profile vectors of the primary gene and the interactor gene, p-value

3. Mutex – Mutual exclusivity of a genetic event E (amplification, deletion or mutation) for the primary gene and the interactor gene; calculated with a hypergeometric test to calculate the probability of co-occurrence of the genetic event E in both genes, in

patient samples (from TCGA). And,

4. SharedPathway The probability of both genes sharing pathways being by chance. Thep-value for the primary gene and the interactor gene haring common pathways, calculated with a hypergeometric test to calculates the probability of co-existence of both genes in pathways annotated in KEGG, Reactome and PID pathway databases.

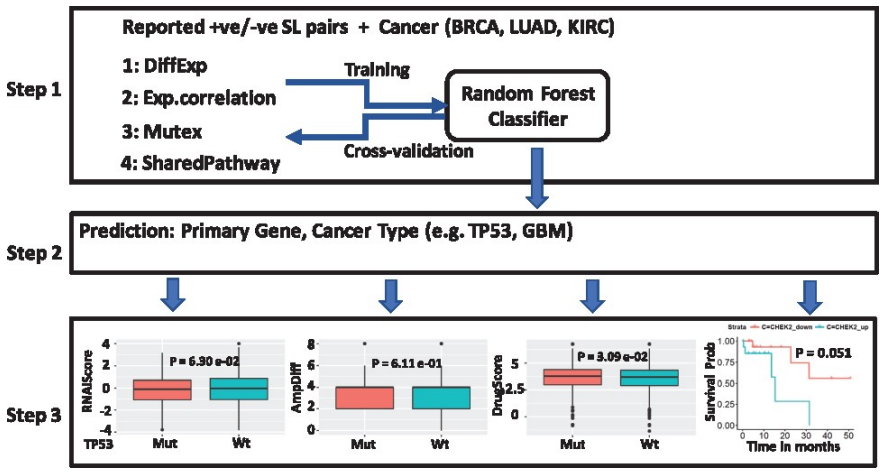


Figure C.1: The DiscoverSL workflow showing the trained random forest, RF model on combined multiple data types (Step 1), applied to new data for prediction (Step 2) and validation (Step 3), Das et al. 2018

According to Das et al (2018),

DiffExp p-value (Differential expression of interactor gene based on mutation status of primary gene)

Using expression profile retrieved from the TCGA dataset, Das et al estimated differential expression of feature read counts using the EdgeR R package[404]. TCGA data consist of processed RNA-Seq data for 9264 tumor and 741 normal samples across 24 cancer types, made available as GEO accession [GSE62944](http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE62
EdgeR normalizes for RNA composition and library size using Trimmed Mean of M-values (TMM) normalization. EdgeR uses quantile-adjusted conditional maximum likelihood (qCML)

method for estimating dispersion which calculates the likelihood by conditioning on the total counts for each tag. An exact test based on the qCML methods is carried on calculating the differential expression of interactor gene between two groups: with and without mutation in the primary gene. Knowing the conditional distribution for the sum of counts in a group, the exact p-values are computed by summing over all sums of counts that have a probability less than the probability under the null hypothesis of the observed sum of counts[399].

Exp.correlation p-value

In each cancer type, Das et al. computes the Pearson's correlation co-efficient between the expression profile vectors of the primary gene and the interactor gene as:

$$r = \frac{\sum_{i=1}^{n}(e1_i - \bar{e1})(e2_i - \bar{e2})}{\sqrt{\sum_{i=1}^{n}(e1_i - \bar{e1})^2}\sqrt{\sum_{i=1}^{n}(e2_i - \bar{e2})^2}} \tag{C.3}$$

Where,

$e1_i$ = Expression of Gene1 in ith sample in cancer type

$e2_i$ = Expression of Gene2 in ith sample in cancer type

$\bar{e1}$ = Mean Expression of Gene1 in all samples in cancer type

$\bar{e2}$ = Mean Expression of Gene2 in all samples in cancer type

n = Number of samples in cancer type

The significance of the correlation r between $e1$ and $e2$ for n number of samples in cancer type is calculated using t-statistics, to test the null hypothesis that the correlation r between $e1$ and $e2$ is coming from a population where the true correlation of $e1$ and $e2$ is zero:

$$t = \frac{r}{\sqrt{(1 - r^2)/(N - 2)}} \tag{C.4}$$

Mutex p-value (Mutual Exclusivity)

Das et al estimated the mutual exclusivity of a genetic event E (amplification, deletion or mutation) for the primary gene (Gene1) and the interactor gene (Gene2) with a hypergeometric test that calculates the probability of co-occurrence of the genetic event E in Gene1 and Gene2 in patient samples (from TCGA) for a specific cancer type.

Given P_{Amp}, P_{Del} and P_{Mut} as the hypergeometric p-values for co-occurrence of the genetic events amplification, deletion and mutation respectively for Gene1 and Gene2, the estimated hypergeometric p-values is given as follows:

$$P_{Amp} = \sum_{i=S_{12amp}}^{\min(S_{1amp},S_{2amp})} \frac{\binom{S_{1amp}}{i}\binom{S_T - S_{1amp}}{S_{2amp}-i}}{\binom{S_T}{S_{2amp}}} \tag{C.5}$$

Where,

S_{12amp} = Number of cancer samples for the cancer type with amplification in both Gene1 and Gene2

S_{1amp} = Number of cancer samples for the cancer type with amplification in Gene1

S_{2amp} = Number of cancer samples for the cancer type with amplification in Gene2

S_T = Total Number of cancer samples for the cancer type

$$P_{Del} = \sum_{i=S_{12del}}^{\min(S_{1del},S_{2del})} \frac{\binom{S_{1del}}{i}\binom{S_T - S_{1del}}{S_{2del}-i}}{\binom{S_T}{S_{2del}}} \tag{C.6}$$

Where,

S_{12del} = Number of cancer samples for the cancer type with deletion in both Gene1 and Gene2

S_{1del} = Number of cancer samples for the cancer type with deletion in Gene1

S_{2del} = Number of cancer samples for the cancer type with deletion in Gene2

S_T = Total Number of cancer samples for the cancer type

$$P_{Mut} = \sum_{i=S_{12mut}}^{\min(S_{1mut},S_{2mut})} \frac{\binom{S_{1mut}}{i}\binom{S_T - S_{1mut}}{S_{2mut}-i}}{\binom{S_T}{S_{2mut}}} \tag{C.7}$$

Where,

S_{12mut} = Number of cancer samples for the cancer type with mutation in both Gene1 and Gene2

S_{1mut} = Number of cancer samples for the cancer type with mutation in Gene1

S_{2mut} = Number of cancer samples for the cancer type with mutation in Gene2

S_T = Total Number of cancer samples for the cancer type

The mutual exclusivity p-values (MutexAmp, MutexDel and MutexMut) representing the p-value for non-co-occurrence of the events of amplification, deletion and mutation was calculated as:

$$\text{Mutex}_{Amp} = 1 - P_{Amp} \tag{C.8}$$

$$\text{Mutex}_{Del} = 1 - P_{Del} \tag{C.9}$$

$$\text{Mutex}_{Mut} = 1 - P_{Mut} \tag{C.10}$$

$$\tag{C.11}$$

The three Mutex p-values were combined into a single p-value using the Fisher's method and corrected for multiple testing using the false discovery rate (FDR) approach.

SharedPathway p-value

The p-value for the primary gene (Gene1) and the interactor gene (Gene2) sharing common pathways was also calculated with a hypergeometric test that calculates the probability of co-existence of Gene1 and Gene2 in pathways annotated in KEGG, Reactome and PID pathway databases – all collected from the Molecular Signatures Database (MSigDB) of Broad Institute [405]. Given $P_{pathway}$ as the hypergeometric p-value for co-existence of Gene1 and Gene2 in common pathways:

$$P_Pathway = \sum_{i=S_{12path}}^{\min(S_{1path},S_{2path})} \frac{\binom{S_{1path}}{i}\binom{S_T - S_{1path}}{S_{2path}-i}}{\binom{S_T}{S_{2path}}} \tag{C.12}$$

Where,

S_{12path} = Number of pathways having both Gene1 and Gene2

S_{1path} = Number of pathways having Gene1

S_{2path} = Number of pathways having Gene2

S_T = Total Number of annotated pathways

The Random Forest Classifier

Das et al's random forest classifier is trained on a curated set of 2130 validated positive and negative SL pairs from siRNA screens and or those reported in literature. These included 1268 positive and 862 negative SL examples. Three cancer types, Breast Invasive Carcinoma (BRCA), Lung Adenocarcinoma (LUAD) and Kidney Renal carcinoma (KIRC) were used to train the model. Two methods of cross-validation; Leave-One-Out and 10-fold cross-validation were used to estimate the predictive performance of the Random Forest model.

Patient survival analyses

According to Das et al., to assess the clinical outcome of under-expression or over-expression of the predicted SL gene (gene2) in cases with mutation in the primary gene (gene1), difference in patient disease-free survival, is calculated using TCGA-provided clinical data. The genes are considered as over-expressed or under-expressed in a sample if their expression is above median or below median (respectively) of its expression in all samples. Kaplan-Meier survival curves were generated and the difference in patient survival is calculated between two groups of samples, gene2 is under-expressed in presence of mutation in gene1 and gene2 is over-expressed in presence of mutation in gene1, to check whether suppressing gene2 in samples carrying mutation in gene1 improves cancer patient survival.

C.2.3 SL-BioDP

SL-BioDP is Deng et al's implementation[400] of Das et al's DiscoverSL pipeline [400], SL-BioDP, Synthetic Lethality Bio Discovery Portal (SL-BioDP) builds and generalizes on the models developed in DiscoverSL, and in addition extends the cancer types incorporated in model to 18 cancer genome atlas cohorts. It bridges the divide between SynLethDB, a collection of synthetic lethal partners from multiple sources and real world clinically relevant data.

C.3 Results and Discussions

C.3.1 SynLethDB

SynLethDB has $35,943$ documented human synthetic lethal interaction with a median score of 0.400 (ave. = 0.390, max. = 1.000, min. = 0.000), from 295 cell lines. A query of features in synthetic lethal relationship to Histone deacetylases, reports 52 features with a median lethality score of 0.455 (ave. = 0.448, max. = 0.740, min. = 0.030). Of these, 25 have lethality score greater or equal to 0.500 (see Table C.2). Amongst the top most

Table C.2: Table of SynLethDB-derived Histone Deacetylases (HDACs) Lethal Partners

	gene_a.name	gene_b.name	SL.pubmed_id	SL.statistic_score
14472	HDAC1	HDAC6	31300006	0.74
144721	HDAC1	HDAC6	31300006	0.74
24393	APC	HDAC1	31300006	0.66
3610	HDAC6	PIK3CA	28319113	0.65
6882	HDAC6	PBRM1	28319113	0.65
9306	HDAC2	SMARCA4	28319113	0.65
17881	HDAC2	VHL	28319113	0.65
18523	HDAC2	MAP2K1	28319113	0.65
21339	HDAC6	KDM5C	28319113	0.65
30912	HDAC6	IGF1R	28319113	0.65
14473	BRCA2	HDAC6	28319113	0.65
15982	BRAF	HDAC2	28319113	0.65
15985	BRCA1	HDAC2	28319113	0.65
24392	CDK4	HDAC1	28319113	0.65
27531	HDAC11	NAE1	23100467	0.629
27654	HDAC10	NAE1	23100467	0.629
27339	HDAC1	NAE1	23100467	0.611
27484	HDAC2	NAE1	23100467	0.611
27792	HDAC9	NAE1	23100467	0.611
7684	KRAS	HDAC5	24104479	0.6
27342	HDAC3	NAE1	23100467	0.575
14852	HDAC9	PLK1	23204129	0.56
31244	EGFR	HDAC9	24052078	0.56
27530	HDAC4	NAE1	23100467	0.5

important features from our fuzzy logic regulatory network, the APC gene is observed to be synthetically lethal with the gene HDAC1 (lethality score = 0.660). The TP53 gene is also reported to be in a synthetic lethal relationship with the HDAC1 and HDAC9 genes, albeit with low lethality scores – 0.300 and 0.178 respectively. It can be observed that up and downstream members of canonical pathways involving top nodes (by node importance score) from our regulatory network are well represented in documented synthetic lethal partners to Histone deacetylases, with high computed lethality scores. These include the genes PIK3CA, MAP2K1, BRAF, KRAS, and EGFR with computed lethality scores 0.650, 0.650, 0.650, 0.600 and 0.560 respectively (Table C.2).

C.3.2 SL-BioDP

On the SL-BioDP portal, we searched for predicted synthetic lethal partners to the eleven histone deacetylases (HDAC1, HDAC2, HDAC3,...HDAC11) across all profiled cancer types in the TCGA. We observed that among all the HDACs, only HDAC1 is included as a primary gene in SL Model and all included samples in the model were from invasive breast cancer. Of the top 20 features by node importance from regulatory network, we found predicted sythetic lethal interactions with the HDAC1 gene involving 18 topranked features (Table C.3). Average estimated lethality score is 0.7062. Maximum observed lethality score of 0.8945 was derived for the relationship between HDAC1 and PARVG (Parvin Gamma, an actin-binding protein). Amongst these, the MAPK1 (ERK) gene interaction with the HDAC1 gene shows the most significant association with patient survival with log rank p-value of 0.0177 (Table C.4, Figure C.2)

C.3.3 The MAPK1 Pathway

Quite siginificant among the list of reported synthetically lethal partners to Histone deacetylase from SynLethDB are the MAPK (mitogen activated protein kinase) pathway members. The biological significance of this may not be overlooked given the added validation of the

Table C.3: DiscoverSL Algorithm-based SL-BioDP Table of Synthetic Lethality Predictions

	SL_Primary_Gene	SL_Interactor_Gene	SL_Score	Pvalue_of_Mutual_Exclusivity
13582	HDAC1	MAPK1	0.3267266	0.0000000
12455	HDAC1	PXN	0.6345493	0.0000000
7344	HDAC1	POLR2D	0.7945974	0.0000000
9129	HDAC1	CTNNB1	0.7350223	0.0000000
6356	HDAC1	LMO7	0.8208745	0.0000000
13381	HDAC1	UBC	0.5417156	0.0000000
13033	HDAC1	GALNT12	0.5975057	0.9926907
13293	HDAC1	SMAD2	0.5600730	0.9781925
13334	HDAC1	TP53	0.5518308	0.0000000
5654	HDAC1	CCNK	0.8348050	0.9926907
4152	HDAC1	GNAZ	0.8603052	0.0000000
3579	HDAC1	TNS1	0.8682505	0.0000000
4384	HDAC1	PSMD13	0.8561722	0.0000000
13270	HDAC1	APC	0.5660398	0.9674235
8859	HDAC1	RIMBP3	0.7428765	0.0000000
1696	HDAC1	PARVG	0.8944539	0.0000000
10339	HDAC1	PTEN	0.7031768	0.0000000
6305	HDAC1	RIMBP3B	0.8220788	0.0000000

association of MAPK1 over- and under-expression, against the backdrop of HDAC1 mutation, to patient survival from prediction report from SL-BioDP analyses. The MAPK1 protein acts downstream of the RAS-BRAF signal, as well as those of the PIK3CA-AKT1 pathways; both of which are very well known pro-survival and proliferation signaling pathways in neoplastic cells. The role of a convergence of these in quickly overriding the growth arrest or apoptotic process initiated by vorinostat can be reasoned.

C.4 Conclusion

There are compelling evidences of the role of the MAPK1 pathway in conferring resistance or sensitivity to histone deacetylase inhibitors. Among other potentially synthetically lethal partners to HDACs, it appears to present one with the most biomedical significance from our observation – further supporting a rationale for including molecules that target its associated pathway in therapy, in combination with vorinostat for colorectal cancer.

Table C.4: Table of Association with Survival p-values. For each predicted synthetic lethal pair, the estimated p-values compares survival outcomes in patients with low interactor gene expression and those with high interactor gene expression – the two groups have mutations in the primary gene.

	SL_Primary_Gene	SL_Interactor_Gene	Pvalue_of_Survival
13582	HDAC1	MAPK1	0.0177
12455	HDAC1	PXN	0.1230
7344	HDAC1	POLR2D	0.2210
9129	HDAC1	CTNNB1	0.3090
6356	HDAC1	LMO7	0.4460
13381	HDAC1	UBC	0.4750
13033	HDAC1	GALNT12	0.4880
13293	HDAC1	SMAD2	0.4880
13334	HDAC1	TP53	0.5190
5654	HDAC1	CCNK	0.5610
4152	HDAC1	GNAZ	0.5830
3579	HDAC1	TNS1	0.6180
4384	HDAC1	PSMD13	0.6790
13270	HDAC1	APC	0.7060
8859	HDAC1	RIMBP3	0.9090
1696	HDAC1	PARVG	0.9970
10339	HDAC1	PTEN	0.9970
6305	HDAC1	RIMBP3B	NA

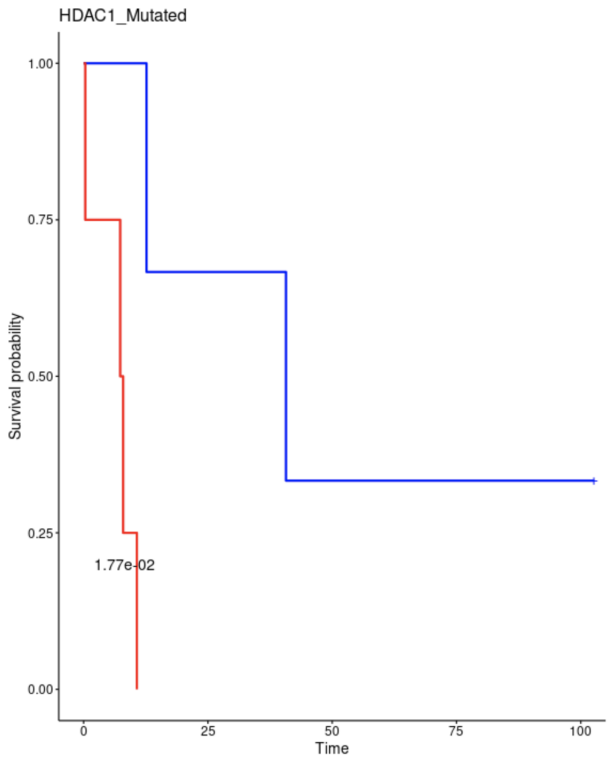


Figure C.2: Kaplan-Meier plot of survival between patients with MAPK1 down (blue) and MAPK1 up (red) expression against a backdrop of HDAC1 mutation in both patient groups.

Bibliography

Bibliography

[1] C. C. Alliance, "colorectal cancer, know the facts," 2021. [Online]. Available: https://www.ccalliance.org/colorectal-cancer-information/facts-and-statistics

[2] A. C. Society, "Key statistics for colorectal cancer," 2021. [Online]. Available: https://www.cancer.org/cancer/colon-rectal-cancer/about/key-statistics.html

[3] R. L. Siegel, K. D. Miller, A. Goding Sauer, S. A. Fedewa, L. F. Butterly, J. C. Anderson, A. Cercek, R. A. Smith, and A. Jemal, "Colorectal cancer statistics, 2020," *CA: a cancer journal for clinicians*, vol. 70, no. 3, pp. 145–164, 2020.

[4] C. for Disease Control and Prevention, "United states cancer statistics: Data visualizations, leading cancer (colon and rectum) cases and deaths, all races/ethnicities, male and female, 2017," 2021. [Online]. Available: https://gis.cdc.gov/Cancer/USCS/DataViz.html

[5] A. C. Society, "Colorectal cancer risk factors," 2021. [Online]. Available: https://www.cancer.org/cancer/colon-rectal-cancer/causes-risks-prevention/risk-factors.html

[6] I. Mármol, C. Sánchez-de Diego, A. Pradilla Dieste, E. Cerrada, and M. J. Rodriguez Yoldi, "Colorectal carcinoma: a general overview and future perspectives in colorectal cancer," *International journal of molecular sciences*, vol. 18, no. 1, p. 197, 2017.

[7] M. G. Fakih, L. Pendyala, G. Fetterly, K. Toth, J. A. Zwiebel, I. Espinoza-Delgado, A. Litwin, Y. M. Rustum, M. E. Ross, J. L. Holleran *et al.*, "A phase i, pharmacokinetic and pharmacodynamic study on vorinostat in combination with 5-fluorouracil, leucovorin, and oxaliplatin in patients with refractory colorectal cancer," *Clinical Cancer Research*, vol. 15, no. 9, pp. 3189–3195, 2009.

[8] M. G. Fakih, "A phase i, pharmacokinetic, and pharmacodynamic study of two schedules of vorinostat in combination with 5-fluorouracil and leucovorin in patients with refractory solid tumors," 2010.

[9] A. H. Ree, S. Dueland, S. Folkvord, K. H. Hole, T. Seierstad, M. Johansen, T. W. Abrahamsen, and K. Flatmark, "Vorinostat, a histone deacetylase inhibitor, combined with pelvic palliative radiotherapy for gastrointestinal carcinoma: the pelvic radiation and vorinostat (pravo) phase 1 study," *The lancet oncology*, vol. 11, no. 5, pp. 459–464, 2010.

[10] M. P. Morelli, J. J. Tentler, G. N. Kulikowski, A.-C. Tan, E. L. Bradshaw-Pierce, T. M. Pitts, A. M. Brown, S. Nallapareddy, J. J. Arcaroli, N. J. Serkova *et al.*, "Preclinical activity of the rational combination of selumetinib (azd6244) in combination with vorinostat in kras-mutant colorectal cancer models," *Clinical Cancer Research*, vol. 18, no. 4, pp. 1051–1062, 2012.

[11] M. Fakih, A. Groman, J. McMahon, G. Wilding, and J. Muindi, "A randomized phase ii study of two doses of vorinostat in combination with 5-fu/lv in patients with refractory colorectal cancer," *Cancer chemotherapy and pharmacology*, vol. 69, no. 3, pp. 743–751, 2012.

[12] J. Vansteenkiste, E. Van Cutsem, H. Dumez, C. Chen, J. L. Ricker, S. S. Randolph, and P. Schöffski, "Early phase ii trial of oral vorinostat in relapsed or refractory breast, colorectal, or non-small cell lung cancer," *Investigational new drugs*, vol. 26, no. 5, pp. 483–488, 2008.

[13] P. M. Wilson, A. El-Khoueiry, S. Iqbal, W. Fazzone, M. J. LaBonte, S. Groshen, D. Yang, K. D. Danenberg, S. Cole, M. Kornacki *et al.*, "A phase i/ii trial of vorinostat in combination with 5-fluorouracil in patients with metastatic colorectal cancer who previously failed 5-fu-based chemotherapy," *Cancer chemotherapy and pharmacology*, vol. 65, no. 5, pp. 979–988, 2010.

[14] D. A. Deming, J. Ninan, H. H. Bailey, J. M. Kolesar, J. Eickhoff, J. M. Reid, M. M. Ames, R. M. McGovern, D. Alberti, R. Marnocha *et al.*, "A phase i study of intermittently dosed vorinostat in combination with bortezomib in patients with advanced solid tumors," *Investigational new drugs*, vol. 32, no. 2, pp. 323–329, 2014.

[15] S. Fu, M. Hou, A. Naing, F. Janku, K. Hess, R. Zinner, V. Subbiah, D. Hong, J. Wheler, S. Piha-Paul *et al.*, "Phase i study of pazopanib and vorinostat: a therapeutic approach for inhibiting mutant p53-mediated angiogenesis and facilitating mutant p53 degradation," *Annals of oncology*, vol. 26, no. 5, pp. 1012–1018, 2015.

[16] D. Mahalingam, M. Mita, J. Sarantopoulos, L. Wood, R. K. Amaravadi, L. E. Davis, A. C. Mita, T. J. Curiel, C. M. Espitia, S. T. Nawrocki *et al.*, "Combined autophagy and hdac inhibition: a phase i safety, tolerability, pharmacokinetic, and pharmacodynamic analysis of hydroxychloroquine in combination with the hdac inhibitor vorinostat in patients with advanced solid tumors," *Autophagy*, vol. 10, no. 8, pp. 1403–1414, 2014.

[17] Y. Wang, F. Janku, S. Piha-Paul, K. Hess, R. Broaddus, L. Liu, N. Shi, M. Overman, S. Kopetz, V. Subbiah *et al.*, "Phase i studies of vorinostat with ixazomib or pazopanib imply a role of antiangiogenesis-based therapy for tp53 mutant malignancies," *Scientific reports*, vol. 10, no. 1, pp. 1–9, 2020.

[18] S. Grant, C. Easley, and P. Kirkpatrick, "Vorinostat," *Nature reviews Drug discovery*, vol. 6, no. 1, pp. 21–22, 2007.

[19] J. C. Stowell, R. I. Huot, and L. Van Voast, "The synthesis of n-hydroxy-n'-phenyloctanediamide and its inhibitory effect on proliferation of axc rat prostate cancer cells," *Journal of medicinal chemistry*, vol. 38, no. 8, pp. 1411–1413, 1995.

[20] V. Richon, Y. Webb, R. Merger, T. Sheppard, B. Jursic, L. Ngo, F. Civoli, R. Breslow, R. Rifkind, and P. Marks, "Second generation hybrid polar compounds are potent inducers of transformed cell differentiation," *Proceedings of the National Academy of Sciences*, vol. 93, no. 12, pp. 5705–5708, 1996.

[21] L. M. Butler, D. B. Agus, H. I. Scher, B. Higgins, A. Rose, C. Cordon-Cardo, H. T. Thaler, R. A. Rifkind, P. A. Marks, and V. M. Richon, "Suberoylanilide hydroxamic acid, an inhibitor of histone deacetylase, suppresses the growth of prostate cancer cells in vitro and in vivo," *Cancer research*, vol. 60, no. 18, pp. 5165–5170, 2000.

[22] L. Huang and A. B. Pardee, "Suberoylanilide hydroxamic acid as a potential therapeutic agent for human breast cancer treatment," *Molecular medicine*, vol. 6, no. 10, pp. 849–866, 2000.

[23] P. N. Munster, T. Troso-Sandoval, N. Rosen, R. Rifkind, P. A. Marks, and V. M. Richon, "The histone deacetylase inhibitor suberoylanilide hydroxamic acid induces differentiation of human breast cancer cells," *Cancer research*, vol. 61, no. 23, pp. 8492–8497, 2001.

[24] A. L. Cooper, V. L. Greenberg, P. S. Lancaster, J. R. van Nagell Jr, S. G. Zimmer, and S. C. Modesitt, "In vitro and in vivo histone deacetylase inhibitor therapy with suberoylanilide hydroxamic acid (saha) and paclitaxel in ovarian cancer," *Gynecologic oncology*, vol. 104, no. 3, pp. 596–601, 2007.

[25] L. M. Krug, T. Curley, L. Schwartz, S. Richardson, P. Marks, J. Chiao, and W. K. Kelly, "Potential role of histone deacetylase inhibitors in mesothelioma: clinical experience with suberoylanilide hydroxamic acid," *Clinical lung cancer*, vol. 7, no. 4, pp. 257–261, 2006.

[26] H. Miles Prince, M. Bishton, and S. Harrison, "The potential of histone deacetylase inhibitors for the treatment of multiple myeloma," *Leukemia & lymphoma*, vol. 49, no. 3, pp. 385–387, 2008.

[27] V. M. Richon, J. Garcia-Vargas, and J. S. Hardwick, "Development of vorinostat: current applications and future perspectives for cancer therapy," *Cancer letters*, vol. 280, no. 2, pp. 201–210, 2009.

[28] S. Claerhout, J. Y. Lim, W. Choi, Y.-Y. Park, K. Kim, S.-B. Kim, J.-S. Lee, G. B. Mills, and J. Y. Cho, "Gene expression signature analysis identifies vorinostat as a candidate therapy for gastric cancer," *PloS one*, vol. 6, no. 9, p. e24662, 2011.

[29] E. Galanis, K. A. Jaeckle, M. J. Maurer, J. M. Reid, M. M. Ames, J. S. Hardwick, J. F. Reilly, A. Loboda, M. Nebozhyn, V. R. Fantin *et al.*, "Phase ii trial of vorinostat in recurrent glioblastoma multiforme: a north central cancer treatment group study," *Journal of clinical oncology*, vol. 27, no. 12, p. 2052, 2009.

[30] C. B. Yoo and P. A. Jones, "Epigenetic therapy of cancer: past, present and future," *Nature reviews Drug discovery*, vol. 5, no. 1, pp. 37–50, 2006.

[31] W. Wolfson, "Epigenetic cancer therapies emerge out of the lab into the limelight," *Chemistry & biology*, vol. 20, no. 4, pp. 455–456, 2013.

[32] M. Wang and H. Lin, "Understanding the function of mammalian sirtuins and protein lysine acylation," *Annual Review of Biochemistry*, vol. 90, 2021.

[33] X. Chai, J. Guo, R. Dong, X. Yang, C. Deng, C. Wei, J. Xu, W. Han, J. Lu, C. Gao *et al.*, "Quantitative acetylome analysis reveals histone modifications that may predict prognosis in hepatitis b-related hepatocellular carcinoma," *Clinical and translational medicine*, vol. 11, no. 3, p. e313, 2021.

[34] J. E. Bolden, M. J. Peart, and R. W. Johnstone, "Anticancer activities of histone deacetylase inhibitors," *Nature reviews Drug discovery*, vol. 5, no. 9, pp. 769–784, 2006.

[35] W. K. Kelly and P. A. Marks, "Drug insight: histone deacetylase inhibitors—development of the new targeted anticancer agent suberoylanilide hydroxamic acid," *Nature Clinical Practice Oncology*, vol. 2, no. 3, pp. 150–157, 2005.

[36] V. M. Richon, T. W. Sandhoff, R. A. Rifkind, and P. A. Marks, "Histone deacetylase inhibitor selectively induces p21waf1 expression and gene-associated histone acetylation," *Proceedings of the National Academy of Sciences*, vol. 97, no. 18, pp. 10 014–10 019, 2000.

[37] C. Yoo, M.-H. Ryu, Y.-S. Na, B.-Y. Ryoo, C.-W. Lee, and Y.-K. Kang, "Vorinostat in combination with capecitabine plus cisplatin as a first-line chemotherapy for patients with metastatic or unresectable gastric cancer: phase ii study and biomarker analysis," *British journal of cancer*, vol. 114, no. 11, pp. 1185–1190, 2016.

[38] Z. Rana, S. Diermeier, M. Hanif, and R. J. Rosengren, "Understanding failure and improving treatment using hdac inhibitors for prostate cancer," *Biomedicines*, vol. 8, no. 2, p. 22, 2020.

[39] K. J. Falkenberg, C. M. Gould, R. W. Johnstone, and K. J. Simpson, "Genome-wide functional genomic and transcriptomic analyses for genes regulating sensitivity to vorinostat," 2014.

[40] K. J. Falkenberg, A. Newbold, C. M. Gould, J. Luu, J. A. Trapani, G. M. Matthews, K. J. Simpson, and R. W. Johnstone, "A genome scale {RNAi} screen identifies {GLI1} as a novel gene regulating vorinostat sensitivity," pp. 1209–1218, 2016.

[41] N. J. O'Neil, M. L. Bailey, and P. Hieter, "Synthetic lethality and cancer," *Nat. Rev. Genet.*, vol. 18, no. 10, pp. 613–623, oct 2017.

[42] V. Marigo, R. L. Johnson, A. Vortkamp, and C. J. Tabin, "Sonic hedgehog differentially regulates expression of gli and gli3 during limb development," *Developmental biology*, vol. 180, no. 1, pp. 273–283, 1996.

[43] A. M. Skoda, D. Simovic, V. Karin, V. Kardum, S. Vranic, and L. Serman, "The role of the hedgehog signaling pathway in cancer: A comprehensive review," *Bosnian journal of basic medical sciences*, vol. 18, no. 1, p. 8, 2018.

[44] M. Niyaz, M. S. Khan, and S. Mudassar, "Hedgehog signaling: an achilles' heel in cancer," *Translational oncology*, vol. 12, no. 10, pp. 1334–1344, 2019.

[45] J. P. Fitch and B. Sokhansanj, "Genomic engineering: moving beyond {DNA} sequence to function," *Proc. IEEE*, vol. 88, no. 12, pp. 1949–1971, 2000.

[46] S. Liang, S. Fuhrman, and R. Somogyi, "Reveal, a general reverse engineering algorithm for inference of genetic network architectures," *Pac. Symp. Biocomput.*, pp. 18–29, 1998.

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

[48] J. Tegner, M. K. S. Yeung, J. Hasty, and J. J. Collins, "Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling," *Proceedings of the National Academy of Sciences*, vol. 100, no. 10, pp. 5944–5949, 2003.

[49] B. A. Sokhansanj, J. P. Fitch, J. N. Quong, and A. A. Quong, "Linear fuzzy gene network models obtained from microarray data by exhaustive search," *BMC Bioinformatics*, vol. 5, p. 108, aug 2004.

[50] P. J. Woolf and Y. Wang, "A fuzzy logic approach to analyzing gene expression data," *Physiol. Genomics*, vol. 3, no. 1, pp. 9–15, jun 2000.

[51] E. Ward, R. L. Sherman, S. Jane Henley, A. Jemal, D. A. Siegel, E. J. Feuer, A. U. Firth, B. A. Kohler, S. Scott, J. Ma, R. N. Anderson, V. Benard, and K. Cronin, "Annual Report to the Nation on the Status of Cancer, 1999–2015, Featuring Cancer in Men and Women ages 20–49," 2019.

[52] H. K. Weir, T. D. Thompson, A. Soman, B. Møller, and S. Leadbetter, "The past, present, and future of cancer incidence in the United States: 1975 through 2020," pp. 1827–1837, 2015.

[53] G. K. Vincent, *The Next Four Decades: The Older Population in the United States : 2010 to 2050*, 2010.

[54] B. D. Smith, G. L. Smith, A. Hurria, G. N. Hortobagyi, and T. A. Buchholz, "Future of cancer incidence in the United States: burdens upon an aging, changing nation," *J. Clin. Oncol.*, vol. 27, no. 17, pp. 2758–2765, jun 2009.

[55] B. K. Edwards, H. L. Howe, L. A. G. Ries, M. J. Thun, H. M. Rosenberg, R. Yancik, P. A. Wingo, A. Jemal, and E. G. Feigal, "Annual report to the nation on the status of cancer, 1973-1999, featuring implications of age and aging on {U.S}. cancer burden," *Cancer*, vol. 94, no. 10, pp. 2766–2792, may 2002.

[56] "Worldwide Cancer Statistics," \url{https://www.cancerresearchuk.org/health-professional/cancer-statistics/worldwide-cancer#heading-Zero}.

[57] "American Cancer Society. Global Cancer Facts & Figures 4th Edition," *Atlanta: American Cancer Society*, 2018.

[58] H. P. J. Buermans and J. T. den Dunnen, "Next generation sequencing technology: Advances and applications," pp. 1932–1941, 2014.

[59] A. McCarthy, "Third generation {DNA} sequencing: pacific biosciences' single molecule real time technology," *Chem. Biol.*, vol. 17, no. 7, pp. 675–676, jul 2010.

[60] Y. Feng, Y. Zhang, C. Ying, D. Wang, and C. Du, "Nanopore-based fourth-generation {DNA} sequencing technology," *Genomics Proteomics Bioinformatics*, vol. 13, no. 1, pp. 4–16, feb 2015.

[61] J. M. Heather and B. Chain, "The sequence of sequencers: The history of sequencing {DNA}," *Genomics*, vol. 107, no. 1, pp. 1–8, jan 2016.

[62] I. H. Consortium and †The International HapMap Consortium, "The International {HapMap} Project," pp. 789–796, 2003.

[63] A. Collins, "{HapMap} Project," 2004.

[64] B. J. Traynor and A. Singleton, "{HapMap} Project," 2008.

[65] C. Holding, "{HapMap} Project launched," pp. spotlight—-20 031 219, 2003.

[66] J. Russell and R. Cohn, *International Hapmap Project*. Book on Demand Limited, apr 2012.

[67] T. E. P. Consortium and The ENCODE Project Consortium, "The {ENCODE} ({ENCyclopedia} Of {DNA} Elements) Project," pp. 636–640, 2004.

[68] ——, "Identification and analysis of functional elements in 1% of the human genome by the {ENCODE} pilot project," pp. 799–816, 2007.

[69] D. J. Thomas, K. R. Rosenbloom, H. Clawson, A. S. Hinrichs, H. Trumbower, B. J. Raney, D. Karolchik, G. P. Barber, R. A. Harte, J. Hillman-Jackson, R. M. Kuhn, B. L. Rhead, K. E. Smith, A. Thakkapallayil, A. S. Zweig, D. Haussler, W. J. Kent, and The ENCODE Project Consortium, "The {ENCODE} Project at {UC} Santa Cruz," pp. D663—-D667, 2007.

[70] T. E. P. Consortium and The ENCODE Project Consortium, "A User's Guide to the Encyclopedia of {DNA} Elements ({ENCODE})," p. e1001046, 2011.

[71] N. de Souza and N. de Souza, "The {ENCODE} project," p. 1046, 2012.

[72] T. E. P. Consortium and The ENCODE Project Consortium, "An integrated encyclopedia of {DNA} elements in the human genome," pp. 57–74, 2012.

[73] S. R. Eddy, "The {ENCODE} project: Missteps overshadowing a success," pp. R259—-R261, 2013.

[74] L. Sastre, "Clinical implications of the {ENCODE} project," pp. 801–802, 2012.

[75] K. Nakai, "Impacts of the {ENCODE} Project," pp. 272–273, 2013.

[76] T. Hampton, "Cancer Genome Atlas," p. 1958, 2006.

[77] B. M. Kuehn, "1000 Genomes Project Promises Closer Look at Variation in Human Genome," p. 2715, 2008.

[78] M. Via, C. Gignoux, and E. Burchard, "The 1000 Genomes Project: new opportunities for research and social challenges," p. 3, 2010.

[79] K. U. Chee-Seng, L. E. Yun, P. Yudi, and C. Kee-Seng, "Whole Genome Resequencing and 1000 Genomes Project," 2010.

[80] M. J. Ellis, M. Gillette, S. A. Carr, A. G. Paulovich, R. D. Smith, K. K. Rodland, R. R. Townsend, C. Kinsinger, M. Mesri, H. Rodriguez, D. C. Liebler, and Clinical Proteomic Tumor Analysis Consortium (CPTAC), "Connecting genomic alterations to cancer biology with proteomics: the {NCI} Clinical Proteomic Tumor Analysis Consortium," *Cancer Discov.*, vol. 3, no. 10, pp. 1108–1112, oct 2013.

[81] J. Ohab, "Developing a 2020 vision for genomics: {NHGRI} launches new round of strategic planning," \url{https://www.genome.gov/news/news-release/Developing-a-2020-vision-for-genomics-NHGRI-launches-new-round-of-strategic-planning}, feb 2018.

[82] E. Voit, *A First Course in Systems Biology*. Garland Science, sep 2017.

[83] W. E. Combs and J. E. Andrews, "Combinatorial rule explosion eliminated by a fuzzy rule configuration," pp. 1–11, 1998.

[84] L. H. Hartwell, P. Szankasi, C. J. Roberts, A. W. Murray, and S. H. Friend, "Integrating genetic approaches into the discovery of anticancer drugs," *Science*, vol. 278, no. 5340, pp. 1064–1068, nov 1997.

[85] W. G. Kaelin Jr and W. G. Kaelin Jr, "The concept of synthetic lethality in the context of anticancer therapy," *Nat. Rev. Cancer*, vol. 5, no. 9, pp. 689–698, sep 2005. [Online]. Available: http://dx.doi.org/10.1038/nrc1691 https://www.ncbi.nlm.nih.gov/pubmed/16110319 http://dx.doi.org/10.1038/nrc1691 LB - KTgh7

[86] H. E. Bryant, N. Schultz, H. D. Thomas, K. M. Parker, D. Flower, E. Lopez, S. Kyle, M. Meuth, N. J. Curtin, and T. Helleday, "Specific killing of {BRCA2-deficient} tumours with inhibitors of {poly(ADP-ribose}) polymerase," *Nature*, vol. 434, no. 7035, pp. 913–917, apr 2005.

[87] H. Farmer, N. McCabe, C. J. Lord, A. N. J. Tutt, D. A. Johnson, T. B. Richardson, M. Santarosa, K. J. Dillon, I. Hickson, C. Knights, N. M. B. Martin, S. P. Jackson, G. C. M. Smith, and A. Ashworth, "Targeting the {DNA} repair defect in {BRCA} mutant cells as a therapeutic strategy," *Nature*, vol. 434, no. 7035, pp. 917–921, apr 2005.

[88] J. Nip, D. K. Strom, B. E. Fee, G. Zambetti, J. L. Cleveland, and S. W. Hiebert, "{E2F-1} cooperates with topoisomerase {II} inhibition and {DNA} damage to selectively augment p53-independent apoptosis," *Mol. Cell. Biol.*, vol. 17, no. 3, pp. 1049–1056, mar 1997.

[89] A. Almasan, Y. Yin, R. E. Kelly, E. Y. Lee, A. Bradley, W. Li, J. R. Bertino, and G. M. Wahl, "Deficiency of retinoblastoma protein leads to inappropriate S-phase

entry, activation of {E2F-responsive} genes, and apoptosis," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 92, no. 12, pp. 5436–5440, jun 1995.

[90] D. Banerjee, B. Schnieders, J. Z. Fu, D. Adhikari, S. C. Zhao, and J. R. Bertino, "Role of {E2F-1} in chemosensitivity," *Cancer Res.*, vol. 58, no. 19, pp. 4292–4296, oct 1998.

[91] S. Dolma, S. L. Lessnick, W. C. Hahn, and B. R. Stockwell, "Identification of genotype-selective antitumor agents using synthetic lethal chemical screening in engineered human tumor cells," *Cancer Cell*, vol. 3, no. 3, pp. 285–296, mar 2003.

[92] J. S. Isaacs, W. Xu, and L. Neckers, "Heat shock protein 90 as a molecular target for cancer therapeutics," *Cancer Cell*, vol. 3, no. 3, pp. 213–217, 2003.

[93] P. Workman, "Altered states: selectively drugging the Hsp90 cancer chaperone," *Trends Mol. Med.*, vol. 10, no. 2, pp. 47–51, feb 2004.

[94] A. L. Goldberg, "Protein degradation and protection against misfolded or damaged proteins," *Nature*, vol. 426, no. 6968, pp. 895–899, 2003.

[95] S. V. Rajkumar, S. Vincent Rajkumar, P. G. Richardson, T. Hideshima, and K. C. Anderson, "Proteasome Inhibition As a Novel Therapeutic Target in Human Cancer," *J. Clin. Oncol.*, vol. 23, no. 3, pp. 630–639, 2005.

[96] I. Bajrami, R. Marlow, M. van de Ven, R. Brough, H. N. Pemberton, J. Frankum, F. Song, R. Rafiq, A. Konde, D. B. Krastev, M. Menon, J. Campbell, A. Gulati, R. Kumar, S. J. Pettitt, M. D. Gurden, M. L. Cardenosa, I. Chong, P. Gazinska, F. Wallberg, E. J. Sawyer, L.-A. Martin, M. Dowsett, S. Linardopoulos, R. Natrajan, C. J. Ryan, P. W. B. Derksen, J. Jonkers, A. N. J. Tutt, A. Ashworth, and C. J. Lord, "{E-Cadherin/ROS1} Inhibitor Synthetic Lethality in Breast Cancer," *Cancer Discov.*, vol. 8, no. 4, pp. 498–515, apr 2018.

[97] E. S. Kroll, K. M. Hyland, P. Hieter, and J. J. Li, "Establishing genetic interactions by a synthetic dosage lethality phenotype," *Genetics*, vol. 143, no. 1, pp. 95–102, may 1996.

[98] W. Megchelenbrink, R. Katzir, X. Lu, E. Ruppin, and R. A. Notebaart, "Synthetic dosage lethality in the human metabolic network is highly predictive of tumor growth and cancer patient survival," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 112, no. 39, pp. 12 217–12 222, sep 2015.

[99] L. H. Hartwell, P. Szankasi, C. J. Roberts, A. W. Murray, and S. H. Friend, "Integrating genetic approaches into the discovery of anticancer drugs," *Science*, vol. 278, no. 5340, pp. 1064–1068, nov 1997.

[100] J. A. Simon, P. Szankasi, D. K. Nguyen, C. Ludlow, H. M. Dunstan, C. J. Roberts, E. L. Jensen, L. H. Hartwell, and S. H. Friend, "Differential toxicities of anticancer agents among {DNA} repair and checkpoint mutants of Saccharomyces cerevisiae," *Cancer Res.*, vol. 60, no. 2, pp. 328–333, jan 2000.

[101] B. R. Stockwell, S. J. Haggarty, and S. L. Schreiber, "High-throughput screening of small molecules in miniaturized mammalian cell-based assays involving post-translational modifications," *Chem. Biol.*, vol. 6, no. 2, pp. 71–83, feb 1999.

[102] C. J. Torrance, V. Agrawal, B. Vogelstein, and K. W. Kinzler, "Use of isogenic human cancer cells for high-throughput screening and drug discovery," *Nat. Biotechnol.*, vol. 19, no. 10, pp. 940–945, oct 2001.

[103] A. Bender and J. R. Pringle, "Use of a screen for synthetic lethal and multicopy suppressee mutants to identify two new genes involved in morphogenesis in Saccharomyces cerevisiae," *Mol. Cell. Biol.*, vol. 11, no. 3, pp. 1295–1305, 1991.

[104] A. Simons, N. Dafni, I. Dotan, Y. Oron, and D. Canaani, "Establishment of a chemical synthetic lethality screen in cultured human cells," *Genome Res.*, vol. 11, no. 2, pp. 266–273, feb 2001.

[105] A. H. Simons, N. Dafni, I. Dotan, Y. Oron, and D. Canaani, "Genetic synthetic lethality screen at the single gene level in cultured human cells," *Nucleic Acids Res.*, vol. 29, no. 20, p. E100, oct 2001.

[106] V. R. Fantin and P. Leder, "F16, a Mitochondriotoxic Compound, Triggers Apoptosis or Necrosis Depending on the Genetic Background of the Target Carcinoma Cell," *Cancer Res.*, vol. 64, no. 1, pp. 329–336, 2004.

[107] V. R. Fantin, M. J. Berardi, L. Scorrano, S. J. Korsmeyer, and P. Leder, "A novel mitochondriotoxic small molecule that selectively inhibits tumor cell growth," *Cancer Cell*, vol. 2, no. 1, pp. 29–42, 2002.

[108] S. Dolma, S. L. Lessnick, W. C. Hahn, and B. R. Stockwell, "Identification of genotype-selective antitumor agents using synthetic lethal chemical screening in engineered human tumor cells," *Cancer Cell*, vol. 3, no. 3, pp. 285–296, mar 2003.

[109] X. Lu and H. Robert Horvitz, "lin-35 and lin-53, Two Genes that Antagonize a C. elegans Ras Pathway, Encode Proteins Similar to Rb and Its Binding Protein {RbAp48}," *Cell*, vol. 95, no. 7, pp. 981–991, 1998.

[110] D. S. Fay, "lin-35/Rb and ubc-18, an {E2} ubiquitin-conjugating enzyme, function redundantly to control pharyngeal morphogenesis in C. elegans," *Development*, vol. 130, no. >14, pp. 3319–3330, 2003.

[111] D. S. Fay, S. Keenan, and M. Han, "fzr-1 and lin-35/Rb function redundantly to control cell proliferation in C. elegans as revealed by a nonbiased synthetic screen," *Genes Dev.*, vol. 16, no. 4, pp. 503–517, 2002.

[112] K. A. Edgar, M. Belvin, A. L. Parks, K. Whittaker, M. B. Mahoney, M. Nicoll, C. C. Park, C. G. Winter, F. Chen, K. Lickteig, F. Ahmad, H. Esengil, M. V. Lorenzi, A. Norton, B. A. Rupnow, L. Shayesteh, M. Tabios, L. M. Young, P. M. Carroll, C. Kopczynski, G. D. Plowman, L. S. Friedman, and H. L. Francis-Lang, "Synthetic lethality of retinoblastoma mutant cells in the Drosophila eye by mutation of a novel peptidyl prolyl isomerase gene," *Genetics*, vol. 170, no. 1, pp. 161–171, may 2005.

[113] R. S. Kamath, A. G. Fraser, Y. Dong, G. Poulin, R. Durbin, M. Gotta, A. Kanapin, N. Le Bot, S. Moreno, M. Sohrmann, D. P. Welchman, P. Zipperlen, and J. Ahringer, "Systematic functional analysis of the Caenorhabditis elegans genome using {RNAi}," *Nature*, vol. 421, no. 6920, pp. 231–237, jan 2003.

[114] K. Ashrafi, F. Y. Chang, J. L. Watts, A. G. Fraser, R. S. Kamath, J. Ahringer, and G. Ruvkun, "Genome-wide {RNAi} analysis of Caenorhabditis elegans fat regulatory genes," *Nature*, vol. 421, no. 6920, pp. 268–272, jan 2003.

[115] S. Cherry, T. Doukas, S. Armknecht, S. Whelan, H. Wang, P. Sarnow, and N. Perrimon, "Genome-wide {RNAi} screen reveals a specific sensitivity of {IRES-containing} {RNA} viruses to host translation inhibition," *Genes Dev.*, vol. 19, no. 4, pp. 445–452, feb 2005.

[116] J.-F. Rual, J. Ceron, J. Koreth, T. Hao, A.-S. Nicot, T. Hirozane-Kishikawa, J. Vandenhaute, S. H. Orkin, D. E. Hill, S. van den Heuvel, and M. Vidal, "Toward improving Caenorhabditis elegans phenome mapping with an {ORFeome-based} {RNAi} library," *Genome Res.*, vol. 14, no. 10B, pp. 2162–2168, oct 2004.

[117] A. T. Willingham, Q. L. Deveraux, G. M. Hampton, and P. Aza-Blanc, "{RNAi} and {HTS}: exploring cancer by systematic loss-of-function," *Oncogene*, vol. 23, no. 51, pp. 8392–8400, 2004.

[118] L. Cong, F. A. Ran, D. Cox, S. Lin, R. Barretto, N. Habib, P. D. Hsu, X. Wu, W. Jiang, L. A. Marraffini, and F. Zhang, "Multiplex genome engineering using {CRISPR/Cas} systems," *Science*, vol. 339, no. 6121, pp. 819–823, feb 2013.

[119] M. Jinek, A. East, A. Cheng, S. Lin, E. Ma, and J. Doudna, "{RNA-programmed} genome editing in human cells," *Elife*, vol. 2, p. e00471, jan 2013.

[120] P. Mali, L. Yang, K. M. Esvelt, J. Aach, M. Guell, J. E. DiCarlo, J. E. Norville, and G. M. Church, "{RNA-guided} human genome engineering via Cas9," *Science*, vol. 339, no. 6121, pp. 823–826, feb 2013.

[121] O. Shalem, N. E. Sanjana, and F. Zhang, "High-throughput functional genomics using {CRISPR–Cas9}," *Nat. Rev. Genet.*, vol. 16, no. 5, pp. 299–311, 2015.

[122] A. J. Aguirre, R. M. Meyers, B. A. Weir, F. Vazquez, C.-Z. Zhang, U. Ben-David, A. Cook, G. Ha, W. F. Harrington, M. B. Doshi, M. Kost-Alimova, S. Gill, H. Xu, L. D. Ali, G. Jiang, S. Pantel, Y. Lee, A. Goodale, A. D. Cherniack, C. Oh, G. Kryukov, G. S. Cowley, L. A. Garraway, K. Stegmaier, C. W. Roberts, T. R. Golub, M. Meyerson, D. E. Root, A. Tsherniak, and W. C. Hahn, "Genomic Copy Number Dictates a {Gene-Independent} Cell Response to {CRISPR/Cas9} Targeting," *Cancer Discov.*, vol. 6, no. 8, pp. 914–929, aug 2016.

[123] D. M. Munoz, P. J. Cassiani, L. Li, E. Billy, J. M. Korn, M. D. Jones, J. Golji, D. A. Ruddy, K. Yu, G. McAllister, A. DeWeck, D. Abramowski, J. Wan, M. D. Shirley, S. Y. Neshat, D. Rakiec, R. de Beaumont, O. Weber, A. Kauffmann, E. R. McDonald 3rd, N. Keen, F. Hofmann, W. R. Sellers, T. Schmelzle, F. Stegmeier, and

M. R. Schlabach, "{CRISPR} Screens Provide a Comprehensive Assessment of Cancer Vulnerabilities but Generate {False-Positive} Hits for Highly Amplified Genomic Regions," *Cancer Discov.*, vol. 6, no. 8, pp. 900–913, aug 2016.

[124] T. Wang, K. Birsoy, N. W. Hughes, K. M. Krupczak, Y. Post, J. J. Wei, E. S. Lander, and D. M. Sabatini, "Identification and characterization of essential genes in the human genome," *Science*, vol. 350, no. 6264, pp. 1096–1101, nov 2015.

[125] T. Hart, M. Chandrashekhar, M. Aregger, Z. Steinhart, K. R. Brown, G. MacLeod, M. Mis, M. Zimmermann, A. Fradet-Turcotte, S. Sun, P. Mero, P. Dirks, S. Sidhu, F. P. Roth, O. S. Rissland, D. Durocher, S. Angers, and J. Moffat, "{High-Resolution} {CRISPR} Screens Reveal Fitness Genes and {Genotype-Specific} Cancer Liabilities," *Cell*, vol. 163, no. 6, pp. 1515–1526, dec 2015.

[126] M. A. Horlbeck, L. A. Gilbert, J. E. Villalta, B. Adamson, R. A. Pak, Y. Chen, A. P. Fields, C. Y. Park, J. E. Corn, M. Kampmann, and J. S. Weissman, "Compact and highly active next-generation libraries for {CRISPR-mediated} gene repression and activation," *Elife*, vol. 5, 2016.

[127] M. Costanzo, A. Baryshnikova, B. VanderSluis, B. Andrews, C. L. Myers, and C. Boone, "Genetic Networks," in *Handbook of Systems Biology*, 2013, pp. 115–135.

[128] A. H. Tong, M. Evangelista, A. B. Parsons, H. Xu, G. D. Bader, N. Pagé, M. Robinson, S. Raghibizadeh, C. W. Hogue, H. Bussey, B. Andrews, M. Tyers, and C. Boone, "Systematic genetic analysis with ordered arrays of yeast deletion mutants," *Science*, vol. 294, no. 5550, pp. 2364–2368, dec 2001.

[129] A. H. Y. Tong, "Global Mapping of the Yeast Genetic Interaction Network," *Science*, vol. 303, no. 5659, pp. 808–813, 2004.

[130] A. Baryshnikova, M. Costanzo, S. Dixon, F. J. Vizeacoumar, C. L. Myers, B. Andrews, and C. Boone, "Synthetic Genetic Array ({SGA}) Analysis in Saccharomyces cerevisiae and Schizosaccharomyces pombe," in *Methods in Enzymology*, 2010, pp. 145–179.

[131] X. Pan, D. S. Yuan, D. Xiang, X. Wang, S. Sookhai-Mahadeo, J. S. Bader, P. Hieter, F. Spencer, and J. D. Boeke, "A robust toolkit for functional profiling of the yeast genome," *Mol. Cell*, vol. 16, no. 3, pp. 487–496, nov 2004.

[132] L. Decourty, C. Saveanu, K. Zemam, F. Hantraye, E. Frachon, J.-C. Rousselle, M. Fromont-Racine, and A. Jacquier, "Linking functionally related genes by sensitive and quantitative characterization of genetic interaction profiles," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 105, no. 15, pp. 5821–5826, apr 2008.

[133] S. R. Collins, K. M. Miller, N. L. Maas, A. Roguev, J. Fillingham, C. S. Chu, M. Schuldiner, M. Gebbia, J. Recht, M. Shales, H. Ding, H. Xu, J. Han, K. Ingvarsdottir, B. Cheng, B. Andrews, C. Boone, S. L. Berger, P. Hieter, Z. Zhang, G. W. Brown, C. James Ingles, A. Emili, C. David Allis, D. P. Toczyski, J. S. Weissman, J. F. Greenblatt, and N. J. Krogan, "Functional dissection of protein complexes involved in yeast chromosome biology using a genetic interaction map," *Nature*, vol. 446, no. 7137, pp. 806–810, 2007.

[134] M. Schuldiner, S. R. Collins, N. J. Thompson, V. Denic, A. Bhamidipati, T. Punna, J. Ihmels, B. Andrews, C. Boone, J. F. Greenblatt, J. S. Weissman, and N. J. Krogan, "Exploration of the function and organization of the yeast early secretory pathway through an epistatic miniarray profile," *Cell*, vol. 123, no. 3, pp. 507–519, nov 2005.

[135] S. L. Ooi, D. D. Shoemaker, and J. D. Boeke, "{DNA} helicase gene interaction network defined using synthetic lethality analyzed by microarray," *Nat. Genet.*, vol. 35, no. 3, pp. 277–286, nov 2003.

[136] ——, "A {DNA} microarray-based genetic screen for nonhomologous end-joining mutants in Saccharomyces cerevisiae," *Science*, vol. 294, no. 5551, pp. 2552–2556, dec 2001.

[137] M. F. La Russa and L. S. Qi, "The New State of the Art: Cas9 for Gene Activation and Repression," *Mol. Cell. Biol.*, vol. 35, no. 22, pp. 3800–3809, nov 2015.

[138] M. D. M. Leiserson, H.-T. Wu, F. Vandin, and B. J. Raphael, "{CoMEt}: a statistical approach to identify combinations of mutually exclusive alterations in cancer," *Genome Biol.*, vol. 16, no. 1, 2015.

[139] Ö. Babur, M. Gönen, B. A. Aksoy, N. Schultz, G. Ciriello, C. Sander, and E. Demir, "Systematic identification of cancer driving signaling pathways based on mutual exclusivity of genomic alterations," *Genome Biol.*, vol. 16, p. 45, feb 2015.

[140] F. Zhang, M. Wu, X.-J. Li, X.-L. Li, C. K. Kwoh, and J. Zheng, "Predicting essential genes and synthetic lethality via influence propagation in signaling pathways of cancer cell fates," *J. Bioinform. Comput. Biol.*, vol. 13, no. 03, p. 1541002, 2015.

[141] M. Wappett, A. Dulak, Z. R. Yang, A. Al-Watban, J. R. Bradford, and J. R. Dry, "Multi-omic measurement of mutually exclusive loss-of-function enriches for candidate synthetic lethal gene pairs," *BMC Genomics*, vol. 17, p. 65, jan 2016.

[142] L. Jerby-Arnon, N. Pfetzer, Y. Y. Waldman, L. McGarry, D. James, E. Shanks, B. Seashore-Ludlow, A. Weinstock, T. Geiger, P. A. Clemons, E. Gottlieb, and E. Ruppin, "Predicting cancer-specific vulnerability via data-driven detection of synthetic lethality," *Cell*, vol. 158, no. 5, pp. 1199–1209, aug 2014.

[143] N. J. O'Neil, M. L. Bailey, and P. Hieter, "Synthetic lethality and cancer," *Nat. Rev. Genet.*, vol. 18, no. 10, pp. 613–623, 2017.

[144] M. Boutros, L. P. Brás, and W. Huber, "Analysis of cell-based {RNAi} screens," *Genome Biol.*, vol. 7, no. 7, p. R66, 2006.

[145] L. Lum, S. Yao, B. Mozer, A. Rovescalli, D. Von Kessler, M. Nirenberg, and P. A. Beachy, "Identification of Hedgehog pathway components by {RNAi} in Drosophila cultured cells," *Science*, vol. 299, no. 5615, pp. 2039–2045, mar 2003.

[146] P. Müller, D. Kuttenkeuler, V. Gesellchen, M. P. Zeidler, and M. Boutros, "Identification of {JAK/STAT} signalling components by genome-wide {RNA} interference," *Nature*, vol. 436, no. 7052, pp. 871–875, aug 2005.

[147] R. DasGupta, A. Kaykas, R. T. Moon, and N. Perrimon, "Functional genomic analysis of the Wnt-wingless signaling pathway," *Science*, vol. 308, no. 5723, pp. 826–833, may 2005.

[148] D. D. Shao, A. Tsherniak, S. Gopal, B. A. Weir, P. Tamayo, N. Stransky, S. E. Schumacher, T. I. Zack, R. Beroukhim, L. A. Garraway, A. A. Margolin, D. E. Root, W. C. Hahn, and J. P. Mesirov, "{ATARiS}: computational quantification of gene suppression phenotypes from multisample {RNAi} screens," *Genome Res.*, vol. 23, no. 4, pp. 665–678, apr 2013.

[149] R. König, C.-Y. Chiang, B. P. Tu, S. F. Yan, P. D. DeJesus, A. Romero, T. Bergauer, A. Orth, U. Krueger, Y. Zhou, and S. K. Chanda, "A probability-based approach for the analysis of large-scale {RNAi} screens," *Nat. Methods*, vol. 4, no. 10, pp. 847–849, oct 2007.

[150] X. D. Zhang, M. Ferrer, A. S. Espeseth, S. D. Marine, E. M. Stec, M. A. Crackower, D. J. Holder, J. F. Heyse, and B. Strulovici, "The Use of Strictly Standardized Mean Difference for Hit Selection in Primary {RNA} Interference {High-Throughput} Screening Experiments," *J. Biomol. Screen.*, vol. 12, no. 4, pp. 497–509, 2007.

[151] X. D. Zhang, F. Santini, R. Lacson, S. D. Marine, Q. Wu, L. Benetti, R. Yang, A. McCampbell, J. P. Berger, D. M. Toolan, E. M. Stec, D. J. Holder, K. A. Soper, J. F. Heyse, and M. Ferrer, "{cSSMD}: assessing collective activity for addressing off-target effects in genome-scale {RNA} interference screens," *Bioinformatics*, vol. 27, no. 20, pp. 2775–2781, oct 2011.

[152] R. E. Bellman, R. E. Kalaba, and Z. L. A, "Abstraction and pattern classification," *RAND Memorandum*, no. RM-4307-PR, 1964.

[153] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[154] ——, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965. [Online]. Available: http://dx.doi.org/10.1016/s0019-9958(65)90241-x LB - 3Lcp

[155] H. T. Nguyen, C. L. Walker, and E. A. Walker, *A First Course in Fuzzy Logic*. CRC Press, Dec. 2018.

[156] P. R. Halmos, *Naive Set Theory*, ser. 0172-6056. Springer-Verlag New York 1974, 2017.

[157] J. Bagaria, *Set Theory, \textit{The Stanford Encyclopedia of Philosophy}*, E. N. Zalta, Ed. Metaphysics Research Lab, Stanford University, 2020.

[158] T. Watkins, "Fuzzy logic: The logic of fuzzy sets," https://www.sjsu.edu/faculty/watkins/fuzzysets.htm, accessed: 2020-2-29.

[159] O. Salazar and J. Soriano, "Convex combination and its application to fuzzy sets and interval-valued fuzzy sets I," pp. 1061–1068, 2015.

[160] F. Clarke, *Functional Analysis, Calculus of Variations and Optimal Control*. Springer Science & Business Media, Feb. 2013.

[161] V. Novák, I. Perfilieva, and J. Močkoř, "Mathematical principles of fuzzy logic," 1999.

[162] P. Tomassi, *Logic*. Routledge, 1999.

[163] L. Goble, *The Blackwell Guide to Philosophical Logic*. Wiley-Blackwell, Aug. 2001.

[164] S. Gottwald, "Many-Valued logics," pp. 675–722, 2007.

[165] M. Bergmann, *An Introduction to Many-Valued and Fuzzy Logic: Semantics, Algebras, and Derivation Systems*. Cambridge University Press, Jan. 2008.

[166] A. Rose, "Systems of logic whose truth-values form lattices," pp. 152–165, 1951.

[167] K. H. Asli, S. A. O. Aliyev, S. Thomas, and D. A. Gopakumar, *Handbook of Research for Fluid and Solid Mechanics: Theory, Simulation, and Experiment*. CRC Press, Nov. 2017.

[168] M. Mares, "Fuzzy sets," p. 2031, 2006.

[169] B. Kosko, "FUZZINESS VS. PROBABILITY," pp. 211–240, 1990.

[170] L. A. Zadeh, *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lotfi A Zadeh*. World Scientific, May 1996.

[171] ——, "The concept of a linguistic variable and its application to approximate reasoning—i," pp. 199–249, 1975.

[172] ——, "Linguistic variables, approximate reasoning and dispositions," *Med. Inform.*, vol. 8, no. 3, pp. 173–186, Jul. 1983.

[173] A. Newell and H. A. Simon, *Human Problem Solving*. Echo Point Books & Media, Feb. 2019.

[174] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—II," pp. 301–357, 1975.

[175] J. P. Fitch and B. Sokhansanj, "Genomic engineering: moving beyond DNA sequence to function," pp. 1949–1971, 2000.

[176] S. Liang, S. Fuhrman, and R. Somogyi, "Reveal, a general reverse engineering algorithm for inference of genetic network architectures," *Pac. Symp. Biocomput.*, pp. 18–29, 1998.

[177] L. Glass and S. A. Kauffman, "The logical analysis of continuous, non-linear biochemical control networks," *J. Theor. Biol.*, vol. 39, no. 1, pp. 103–129, Apr. 1973.

[178] J. Tegner, M. K. S. Yeung, J. Hasty, and J. J. Collins, "Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 100, no. 10, pp. 5944–5949, May 2003.

[179] B. A. Sokhansanj, J. P. Fitch, J. N. Quong, and A. A. Quong, "Linear fuzzy gene network models obtained from microarray data by exhaustive search," *BMC Bioinformatics*, vol. 5, p. 108, Aug. 2004.

[180] P. J. Woolf and Y. Wang, "A fuzzy logic approach to analyzing gene expression data," *Physiol. Genomics*, vol. 3, no. 1, pp. 9–15, Jun. 2000.

[181] K. Raza, "Fuzzy logic based approaches for gene regulatory network inference," *Artif. Intell. Med.*, vol. 97, pp. 189–203, Jun. 2019.

[182] M. Gormley, V. U. Akella, J. N. Quong, and A. A. Quong, "An integrated framework to model cellular phenotype as a component of biochemical networks," *Adv. Bioinformatics*, vol. 2011, p. 608295, Nov. 2011.

[183] W. Van Leekwijck and E. E. Kerre, "Defuzzification: criteria and classification," pp. 159–178, 1999.

[184] D. P. Filev and R. R. Yager, "A generalized defuzzification method via bad distributions," pp. 687–697, 1991.

[185] T. A. Al Qazlan, A. Hamdi-Cherif, and C. Kara-Mohamed, "State of the art of fuzzy methods for gene regulatory networks inference," *ScientificWorldJournal*, vol. 2015, p. 148010, Mar. 2015.

[186] R. Reynolds, "Gene expression data analysis using fuzzy logic," Master's thesis, The University of Maine, 2001.

[187] H. Ressom, R. Reynolds, and R. S. Varghese, "Increasing the efficiency of fuzzy logic-based gene expression data analysis," *Physiol. Genomics*, vol. 13, no. 2, pp. 107–117, Apr. 2003.

[188] R. Ram, M. Chetty, and T. I. Dix, "Fuzzy model for gene regulatory network," *2006 IEEE International Conference on Evolutionary Computation*, 2006.

[189] W. E. Combs and J. E. Andrews, "Combinatorial rule explosion eliminated by a fuzzy rule configuration," pp. 1–11, 1998.

[190] W. Combs, "Reconfiguring the fuzzy rule matrix for large time-critical applications,," in *3rd Annual Int. Conf. Fuzzy-Neural Applicat., Systems, Tools*, Nov. 1995, pp. 18:1–18:7.

[191] B. A. Sokhansanj, J. B. Garnham, and J. Patrick Fitch, "Interpreting microarray data to build models of microbial genetic regulation networks," 2002.

[192] B. A. Sokhansanj and J. P. Fitch, "URC fuzzy modeling and simulation of gene regulation," 2001.

[193] H. Liu and H. Motoda, "Feature selection for knowledge discovery and data mining," 1998.

[194] ——, *Computational Methods of Feature Selection*. CRC Press, Oct. 2007.

[195] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[196] R. Kohavi and G. H. John, "Wrappers for feature subset selection," pp. 273–324, 1997.

[197] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," pp. 245–271, 1997.

[198] S. H. Huang, "Supervised feature selection: A tutorial," 2015.

[199] L. Haar, K. Anding, K. Trambitckii, and G. Notni, "Comparison between supervised and unsupervised feature selection methods," 2019.

[200] E. B. Fowlkes, R. Gnanadesikan, and J. R. Kettenring, "Variable selection in clustering," pp. 205–228, 1988.

[201] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," *Journal of Machine Learning Research*, no. 5, pp. 845–889, 2004.

[202] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *J. Mach. Learn. Res.*, vol. 5, pp. 1205–1224, 2004.

[203] F. M. Lopes, D. C. Martins-Jr, J. Barrera, and R. M. Cesar-Jr, "An iterative feature selection method for GRNs inference by exploring topological properties," *arXiv:1107.5000v1*, Jul. 2011.

[204] Foldiak and Foldiak, "Adaptive network for optimal linear feature extraction," 1989.

[205] F. M. Lopes, D. C. Martins, J. Barrera, and R. M. Cesar, "SFFS-MR: A floating search strategy for GRNs inference," pp. 407–418, 2010.

[206] S. Solorio-Fernández, J. Ariel Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A review of unsupervised feature selection methods," pp. 907–948, 2020.

[207] J. H. Gennari, P. Langley, and D. Fisher, "Models of incremental concept formation," pp. 11–61, 1989.

[208] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," pp. 121–129, 1994.

[209] H. Almuallim and T. G. Dietterich, "Learning boolean concepts in the presence of many irrelevant features," pp. 279–305, 1994.

[210] ——, "Learning with many irrelevant features," *AAAI-91 Proceedings*, 1991.

[211] J. M. Stuart, E. Segal, D. Koller, and S. K. Kim, "A gene-coexpression network for global discovery of conserved genetic modules," *Science*, vol. 302, no. 5643, pp. 249–255, Oct. 2003.

[212] R. F. Hashimoto, S. Kim, I. Shmulevich, W. Zhang, M. L. Bittner, and E. R. Dougherty, "Growing genetic regulatory networks from seed genes," *Bioinformatics*, vol. 20, no. 8, pp. 1241–1247, May 2004.

[213] E. R. Dougherty, M. Brun, J. M. Trent, and M. L. Bittner, "Conditioning-based modeling of contextual genomic regulation," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 6, no. 2, pp. 310–320, Apr. 2009.

[214] N. Ghaffari, I. Ivanov, X. Qian, and E. R. Dougherty, "A CoD-based reduction algorithm for designing stationary control policies on boolean networks," *Bioinformatics*, vol. 26, no. 12, pp. 1556–1563, Jun. 2010.

[215] A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. D. Favera, and A. Califano, "ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context," 2006.

[216] J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins, and T. S. Gardner, "Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles," *PLoS Biol.*, vol. 5, no. 1, p. e8, Jan. 2007.

[217] A. Rao, A. O. Hero, 3rd, D. J. States, and J. D. Engel, "Using directed information to build biologically relevant influence networks," *J. Bioinform. Comput. Biol.*, vol. 6, no. 3, pp. 493–519, Jun. 2008.

[218] W. Zhao, E. Serpedin, and E. R. Dougherty, "Inferring connectivity of genetic regulatory networks using information-theoretic criteria," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 5, no. 2, pp. 262–274, Apr. 2008.

[219] J. Barrera, R. M. Cesar, D. C. Martins, R. Z. N. Vêncio, E. F. Merino, M. M. Yamamoto, F. G. Leonardi, C. A. de B. Pereira, and H. A. del Portillo, "Constructing probabilistic genetic networks of plasmodium falciparum from dynamical expression signals of the intraerythrocytic development cycle," pp. 11–26, 2007.

[220] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression trees," p. 874, 1984.

[221] V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts, "Inferring regulatory networks from expression data using tree-based methods," *PLoS One*, vol. 5, no. 9, Sep. 2010.

[222] V. A. Huynh-Thu and P. Geurts, "dynGENIE3: dynamical GENIE3 for the inference of gene networks from time series expression data," *Sci. Rep.*, vol. 8, no. 1, p. 3384, Feb. 2018.

[223] R. Clausius, *The Mechanical Theory of Heat*, 1879.

[224] L. Boltzmann, "About the relationships and a general mechanical theorem on the main theorem of thermodynamics," *Math-Naturwissenschaften*, vol. 75, pp. 67–73, 1877.

[225] F. M. Lopes, E. A. de Oliveira, and R. M. Cesar, Jr, "Inference of gene regulatory networks from time series by tsallis entropy," *BMC Syst. Biol.*, vol. 5, p. 61, May 2011.

[226] T. M. Cover and J. A. Thomas, "Elements of information theory," 1991.

[227] R. M. Gray, "Entropy and information theory," 1990.

[228] C. H. Wiggins and I. Nemenman, "Process pathway inference via time series analysis," pp. 361–370, 2003.

[229] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," pp. 1119–1125, 1994.

[230] T. Marill and D. Green, "On the effectiveness of receptors in recognition systems," pp. 11–17, 1963.

[231] A. W. Whitney, "A direct method of nonparametric measurement selection," pp. 1100–1103, 1971.

[232] L. d. F. Costa, L. da F. Costa, F. A. Rodrigues, and A. S. Cristino, "Complex networks: the key to systems biology," pp. 591–601, 2008.

[233] H. Jeong, B. Tombort, R. Albert, L. N. Oltvai, and A. L. BarabÃ¾si, "The large-scale organization of metabolic networks," 2011.

[234] N. Guelzim, S. Bottani, P. Bourgine, and F. Képès, "Topological and causal structure of the yeast transcriptional regulatory network," *Nat. Genet.*, vol. 31, no. 1, pp. 60–63, May 2002.

[235] I. Farkas, H. Jeong, T. Vicsek, A. L. Barabási, and Z. N. Oltvai, "The topology of the transcription regulatory network in the yeast, saccharomyces cerevisiae," pp. 601–612, 2003.

[236] A.-L. Barabási and Z. N. Oltvai, "Network biology: understanding the cell's functional organization," pp. 101–113, 2004.

[237] R. Albert, "Scale-free networks in cell biology," pp. 4947–4957, 2005.

[238] A.-L. Barabási, "Scale-free networks: a decade and beyond," *Science*, vol. 325, no. 5939, pp. 412–413, Jul. 2009.

[239] E. T. Jaynes, "Information theory and statistical mechanics," pp. 620–630, 1957.

[240] A. A. Margolin, K. Wang, A. Califano, and I. Nemenman, "Multivariate dependence and genetic networks inference," *IET Syst. Biol.*, vol. 4, no. 6, pp. 428–440, Nov. 2010.

[241] I. Nemenman, "Information theory, multivariate dependence, and genetic network inference," *https://arxiv.org/abs/q-bio/0406015v1*, 2004.

[242] J. Olczak, N. A. Kiani, H. Zenil, and J. Tegner, "Topological evaluation of methods for reconstruction of genetic regulatory networks," 2015.

[243] A. V. Werhli, M. Grzegorczyk, and D. Husmeier, "Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and bayesian networks," *Bioinformatics*, vol. 22, no. 20, pp. 2523–2531, Oct. 2006.

[244] D. Edwards, "Introduction to graphical modelling," 2000.

[245] J. Wang, O. Myklebost, and E. Hovig, "MGraph: graphical models for microarray data analysis," *Bioinformatics*, vol. 19, no. 17, pp. 2210–2211, Nov. 2003.

[246] R. Varshavsky, A. Gottlieb, M. Linial, and D. Horn, "Novel unsupervised feature filtering of biological data," *Bioinformatics*, vol. 22, no. 14, pp. e507–13, Jul. 2006.

[247] P. J. A. Cock, C. J. Fields, N. Goto, M. L. Heuer, and P. M. Rice, "The Sanger {FASTQ} file format for sequences with quality scores, and the {Solexa/Illumina} {FASTQ} variants," *Nucleic Acids Res.*, vol. 38, no. 6, pp. 1767–1771, apr 2010.

[248] B. Rauscher, F. Heigwer, M. Breinig, J. Winter, and M. Boutros, "{GenomeCRISPR} - a database for high-throughput {CRISPR/Cas9} screens," *Nucleic Acids Res.*, vol. 45, no. D1, pp. D679—-D686, jan 2017.

[249] E. E. Schmidt, O. Pelz, S. Buhlmann, G. Kerr, T. Horn, and M. Boutros, "{GenomeRNAi}: a database for cell-based and in vivo {RNAi} phenotypes, 2013 update," *Nucleic Acids Res.*, vol. 41, no. Database issue, pp. D1021—-6, jan 2013.

[250] M. Gilsdorf, T. Horn, Z. Arziman, O. Pelz, E. Kiner, and M. Boutros, "{GenomeRNAi}: a database for cell-based {RNAi} phenotypes. 2009 update," pp. D448—-D452, 2010.

[251] T. Horn, Z. Arziman, J. Berger, and M. Boutros, "{GenomeRNAi}: a database for cell-based {RNAi} phenotypes," pp. D492—-D497, 2007.

[252] G. S. Cowley, B. A. Weir, F. Vazquez, P. Tamayo, J. A. Scott, S. Rusin, A. East-Seletsky, L. D. Ali, W. F. Gerath, S. E. Pantel, P. H. Lizotte, G. Jiang, J. Hsiao, A. Tsherniak, E. Dwinell, S. Aoyama, M. Okamoto, W. Harrington, E. Gelfand, T. M. Green, M. J. Tomko, S. Gopal, T. C. Wong, H. Li, S. Howell, N. Stransky, T. Liefeld, D. Jang, J. Bistline, B. Hill Meyers, S. A. Armstrong, K. C. Anderson, K. Stegmaier, M. Reich, D. Pellman, J. S. Boehm, J. P. Mesirov, T. R. Golub, D. E. Root, and W. C. Hahn, "Parallel genome-scale loss of function screens in 216 cancer cell lines for the identification of context-specific genetic dependencies," *Sci Data*, vol. 1, p. 140035, sep 2014.

[253] A. Tsherniak, F. Vazquez, P. G. Montgomery, B. A. Weir, G. Kryukov, G. S. Cowley, S. Gill, W. F. Harrington, S. Pantel, J. M. Krill-Burger, R. M. Meyers, L. Ali, A. Goodale, Y. Lee, G. Jiang, J. Hsiao, W. F. J. Gerath, S. Howell, E. Merkel, M. Ghandi, L. A. Garraway, D. E. Root, T. R. Golub, J. S. Boehm, and W. C. Hahn, "Defining a Cancer Dependency Map," *Cell*, vol. 170, no. 3, pp. 564—-576.e16, jul 2017.

[254] R. Edgar, M. Domrachev, and A. E. Lash, "Gene expression omnibus: Ncbi gene expression and hybridization array data repository," *Nucleic acids research*, vol. 30, no. 1, pp. 207–210, 2002.

[255] T. Barrett, S. E. Wilhite, P. Ledoux, C. Evangelista, I. F. Kim, M. Tomashevsky, K. A. Marshall, K. H. Phillippy, P. M. Sherman, M. Holko *et al.*, "Ncbi geo: archive for functional genomics data sets—update," *Nucleic acids research*, vol. 41, no. D1, pp. D991–D995, 2012.

[256] E. Clough and T. Barrett, "The Gene Expression Omnibus Database," *Methods in Molecular Biology*, pp. 93–110, 2016.

[257] R. Leinonen, H. Sugawara, M. Shumway, and on behalf of the International Nucleotide Sequence Database Collaboration, "The Sequence Read Archive," pp. D19—-D21, 2011.

[258] Y. Kodama, M. Shumway, R. Leinonen, and on behalf of the International Nucleotide Sequence Database Collaboration, "The sequence read archive: explosive growth of sequencing data," pp. D54—-D56, 2012.

[259] A. Hamosh, A. F. Scott, J. S. Amberger, C. A. Bocchini, and V. A. McKusick, "Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders," *Nucleic acids research*, vol. 33, no. suppl_1, pp. D514–D517, 2005.

[260] J. S. Amberger, C. A. Bocchini, A. F. Scott, and A. Hamosh, "Omim. org: leveraging knowledge across phenotype–gene relationships," *Nucleic acids research*, vol. 47, no. D1, pp. D1038–D1043, 2019.

[261] C. Isella, L. Cantini, S. E. Bellomo, and E. Medico, *TCGAcrcmRNA: TCGA CRC 450 mRNA dataset*, 2020, r package version 1.10.0.

[262] M. Reimers and V. J. Carey, "[8] Bioconductor: An Open Source Framework for Bioinformatics and Computational Biology," pp. 119–134, 2006.

[263] C. G. A. Network *et al.*, "Comprehensive molecular characterization of human colon and rectal cancer," *Nature*, vol. 487, no. 7407, p. 330, 2012.

[264] R. L. Grossman, A. P. Heath, V. Ferretti, H. E. Varmus, D. R. Lowy, W. A. Kibbe, and L. M. Staudt, "Toward a shared vision for cancer genomic data," *New England Journal of Medicine*, vol. 375, no. 12, pp. 1109–1112, 2016.

[265] C. Printz, "Genomic data commons ushers in new era for information sharing," *Cancer*, vol. 122, no. 18, pp. 2777–2778, 2016.

[266] M. A. Jensen, V. Ferretti, R. L. Grossman, and L. M. Staudt, "The nci genomic data commons as an engine for precision medicine," *Blood*, vol. 130, no. 4, pp. 453–459, 2017.

[267] Z. Zhang, K. Hernandez, J. Savage, S. Li, D. Miller, S. Agrawal, F. Ortuno, L. M. Staudt, A. Heath, and R. L. Grossman, "Uniform genomic data analysis in the nci genomic data commons," *Nature communications*, vol. 12, no. 1, pp. 1–11, 2021.

[268] A. Kassambara, "fastqcr: Quality Control of Sequencing Data. {R} package version 0.1.2," 2019.

[269] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2021. [Online]. Available: https://www.R-project.org/

[270] W. de Souza, W. de Souza, B. de Sá Carvalho, and I. Lopes-Cendes, "Rqc: A Bioconductor Package for Quality Control of {High-Throughput} Sequencing Data," 2018.

[271] S. Andrews, "{FastQC}: A quality control tool for high throughput sequence data," \url{https://www.bioinformatics.babraham.ac.uk/projects/fastqc/}.

[272] R. Patro, G. Duggal, M. I. Love, R. A. Irizarry, and C. Kingsford, "Salmon provides fast and bias-aware quantification of transcript expression," *Nat. Methods*, vol. 14, no. 4, pp. 417–419, apr 2017.

[273] R. Patro, S. M. Mount, and C. Kingsford, "Sailfish enables alignment-free isoform quantification from {RNA-seq} reads using lightweight algorithms," *Nat. Biotechnol.*, vol. 32, no. 5, pp. 462–464, may 2014.

[274] N. L. Bray, H. Pimentel, P. Melsted, and L. Pachter, "Near-optimal probabilistic {RNA-seq} quantification," *Nat. Biotechnol.*, vol. 34, no. 5, pp. 525–527, may 2016.

[275] D. C. Wu, J. Yao, K. S. Ho, A. M. Lambowitz, and C. O. Wilke, "Limitations of alignment-free tools in total {RNA-seq} quantification," *BMC Genomics*, vol. 19, no. 1, p. 510, jul 2018.

[276] D. Kim, G. Pertea, C. Trapnell, H. Pimentel, R. Kelley, and S. L. Salzberg, "{TopHat2}: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions," *Genome Biol.*, vol. 14, no. 4, p. R36, apr 2013.

[277] C. Trapnell, L. Pachter, and S. L. Salzberg, "{TopHat}: discovering splice junctions with {RNA-Seq}," *Bioinformatics*, vol. 25, no. 9, pp. 1105–1111, may 2009.

[278] C. Trapnell, B. A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. J. van Baren, S. L. Salzberg, B. J. Wold, and L. Pachter, "Transcript assembly and quantification by {RNA-Seq} reveals unannotated transcripts and isoform switching during cell differentiation," *Nat. Biotechnol.*, vol. 28, no. 5, pp. 511–515, may 2010.

[279] B. Langmead, C. Trapnell, M. Pop, and S. L. Salzberg, "Ultrafast and memory-efficient alignment of short {DNA} sequences to the human genome," *Genome Biol.*, vol. 10, no. 3, p. R25, mar 2009.

[280] "iGenomes {Ready-To-Use} Reference Sequences and Annotations," \url{https://support.illumina.com/sequencing/sequencing_software/igenome.html}.

[281] The SAM/BAM Format Specification Working Group, "Sequence {Alignment/Map} Format Specification."

[282] Y. Liao, G. K. Smyth, and W. Shi, "The {R} package Rsubread is easier, faster, cheaper and better for alignment and quantification of {RNA} sequencing reads," *Nucleic Acids Research*, vol. 47, no. 8, p. e47, may 2019.

[283] M. I. Love, W. Huber, and S. Anders, "Moderated estimation of fold change and dispersion for {RNA-seq} data with {DESeq2}," *Genome Biol.*, vol. 15, no. 12, p. 550, 2014.

[284] S. Anders and W. Huber, "Differential expression analysis for sequence count data," *Nature Precedings*, pp. 1–1, 2010.

[285] M. I. Love, W. Huber, and S. Anders, "Moderated estimation of fold change and dispersion for rna-seq data with deseq2," *Genome biology*, vol. 15, no. 12, pp. 1–21, 2014.

[286] S. Sun, M. Hood, L. Scott, Q. Peng, S. Mukherjee, J. Tung, and X. Zhou, "Differential expression analysis for rnaseq using poisson mixed models," *Nucleic acids research*, vol. 45, no. 11, pp. e106–e106, 2017.

[287] H. Liu, F. Zhang, S. K. Mishra, S. Zhou, and J. Zheng, "Knowledge-guided fuzzy logic modeling to infer cellular signaling networks from proteomic data," *Scientific reports*, vol. 6, no. 1, pp. 1–12, 2016.

[288] C. v. Mering, M. Huynen, D. Jaeggi, S. Schmidt, P. Bork, and B. Snel, "String: a database of predicted functional associations between proteins," *Nucleic acids research*, vol. 31, no. 1, pp. 258–261, 2003.

[289] D. Szklarczyk, A. Franceschini, M. Kuhn, M. Simonovic, A. Roth, P. Minguez, T. Doerks, M. Stark, J. Muller, P. Bork *et al.*, "The string database in 2011: functional interaction networks of proteins, globally integrated and scored," *Nucleic acids research*, vol. 39, no. suppl_1, pp. D561–D568, 2010.

[290] D. Szklarczyk, A. L. Gable, K. C. Nastou, D. Lyon, R. Kirsch, S. Pyysalo, N. T. Doncheva, M. Legeay, T. Fang, P. Bork *et al.*, "The string database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets," *Nucleic Acids Research*, vol. 49, no. D1, pp. D605–D612, 2021.

[291] K. Raza, "Fuzzy logic based approaches for gene regulatory network inference," *Artif. Intell. Med.*, vol. 97, pp. 189–203, jun 2019.

[292] B. A. Sokhansanj, J. B. Garnham, and J. Patrick Fitch, "Interpreting microarray data to build models of microbial genetic regulation networks," 2002. [Online]. Available: http://dx.doi.org/10.1117/12.469450 LB - uq4B

[293] B. A. Sokhansanj and J. P. Fitch, "{URC} fuzzy modeling and simulation of gene regulation."

[294] M. Gormley, V. U. Akella, J. N. Quong, and A. A. Quong, "An integrated framework to model cellular phenotype as a component of biochemical networks," *Adv. Bioinformatics*, vol. 2011, p. 608295, nov 2011.

[295] S. Datta and B. A. Sokhansanj, "Accelerated search for biomolecular network models to interpret high-throughput experimental data," *BMC Bioinformatics*, vol. 8, p. 258, Jul. 2007.

[296] J. M. Mendel, "Fuzzy logic systems for engineering: a tutorial," pp. 345–377, 1995.

[297] C. Anderson and W. Ray, "Improved maximum likelihood estimators for the gamma distribution," *Communications in Statistics-Theory and Methods*, vol. 4, no. 5, pp. 437–448, 1975.

[298] C. Forbes, M. Evans, N. Hastings, and B. Peacock, *Statistical distributions*. John Wiley & Sons, 2011.

[299] H. Kulkarni and S. Powar, "A new method for interval estimation of the mean of the gamma distribution," *Lifetime data analysis*, vol. 16, no. 3, pp. 431–447, 2010.

[300] A. Singh, A. K. Singh, and R. J. Iaci, "Estimation of the exposure point concentration term using a gamma distribution," in *In USEPA, Ed.* Citeseer, 2002.

[301] J. H. Ahrens and U. Dieter, "Generating gamma variates by a modified rejection technique," *Communications of the ACM*, vol. 25, no. 1, pp. 47–54, 1982.

[302] ——, "Computer methods for sampling from gamma, beta, poisson and bionomial distributions," *Computing*, vol. 12, no. 3, pp. 223–246, 1974.

[303] J. Zhang, W. Zhu, Q. Wang, J. Gu, L. F. Huang, and X. Sun, "Differential regulatory network-based quantification and prioritization of key genes underlying cancer drug resistance based on time-course rna-seq data," *PLoS computational biology*, vol. 15, no. 11, p. e1007435, 2019.

[304] A. E. Teschendorff and S. Severini, "Increased entropy of signal transduction in the cancer metastasis phenotype," *BMC systems biology*, vol. 4, no. 1, pp. 1–15, 2010.

[305] D. Ortiz-Arroyo and D. A. Hussain, "An information theory approach to identify sets of key players," in *European Conference on Intelligence and Security Informatics*. Springer, 2008, pp. 15–26.

[306] M. Jalili, A. Salehzadeh-Yazdi, Y. Asgari, S. S. Arab, M. Yaghmaie, A. Ghavamzadeh, and K. Alimoghaddam, "Centiserver: a comprehensive resource, web-based application and r package for centrality analysis," *PloS one*, vol. 10, no. 11, p. e0143111, 2015.

[307] C. Shannon, "A mathematical theory of communication, bell systems technol," *J*, vol. 27, no. 3, pp. 379–423, 1948.

[308] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *Journal of the American statistical association*, vol. 53, no. 282, pp. 457–481, 1958.

[309] M. K. Goel, P. Khanna, and J. Kishore, "Understanding survival analysis: Kaplan-meier estimate," *International journal of Ayurveda research*, vol. 1, no. 4, p. 274, 2010.

[310] J. T. Rich, J. G. Neely, R. C. Paniello, C. C. Voelker, B. Nussenbaum, and E. W. Wang, "A practical guide to understanding kaplan-meier curves," *Otolaryngology—Head and Neck Surgery*, vol. 143, no. 3, pp. 331–336, 2010.

[311] D. R. Cox, "Regression models and life-tables," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 34, no. 2, pp. 187–202, 1972.

[312] J. M. Bland and D. G. Altman, "The logrank test," *Bmj*, vol. 328, no. 7447, p. 1073, 2004.

[313] R. P. Kruzelock and W. Short, "Colorectal cancer therapeutics and the challenges of applied pharmacogenomics." *Current problems in cancer*, vol. 31, no. 5, pp. 315–366, 2007.

[314] S. Lemery, P. Keegan, and R. Pazdur, "First fda approval agnostic of cancer site-when a biomarker defines the indication," *The New England journal of medicine*, vol. 377, no. 15, pp. 1409–1412, 2017.

[315] P. A. T. E. Board, "Colon cancer treatment (pdq®): Health professional version," *PDQ Cancer Information Summaries, originally published by the National Cancer Institute [Internet]*, 2021.

[316] N. J. Llosa, M. Cruise, A. Tam, E. C. Wicks, E. M. Hechenbleikner, J. M. Taube, R. L. Blosser, H. Fan, H. Wang, B. S. Luber *et al.*, "The vigorous immune microenvironment of microsatellite instable colon cancer is balanced by multiple counter-inhibitory checkpoints," *Cancer discovery*, vol. 5, no. 1, pp. 43–51, 2015.

[317] J. C. Dudley, M.-T. Lin, D. T. Le, and J. R. Eshleman, "Microsatellite instability as a biomarker for pd-1 blockade," *Clinical Cancer Research*, vol. 22, no. 4, pp. 813–820, 2016.

[318] J. LaBaer, M. D. Garrett, L. F. Stevenson, J. M. Slingerland, C. Sandhu, H. S. Chou, A. Fattaey, and E. Harlow, "New functional activities for the p21 family of cdk inhibitors." *Genes & development*, vol. 11, no. 7, pp. 847–862, 1997.

[319] R. W. Johnstone and J. D. Licht, "Histone deacetylase inhibitors in cancer therapy: is transcription the primary target?" *Cancer cell*, vol. 4, no. 1, pp. 13–18, 2003.

[320] P. A. Marks, V. M. Richon, T. Miller, and W. K. Kelly, "Histone deacetylase inhibitors." *Advances in cancer research*, vol. 91, pp. 137–168, 2004.

[321] A. Maloverjan, M. Piirsoo, P. Michelson, P. Kogerman, and T. Østerlund, "Identification of a novel serine/threonine kinase ulk3 as a positive regulator of hedgehog pathway," *Experimental cell research*, vol. 316, no. 4, pp. 627–637, 2010.

[322] M. Murone, S.-M. Luoh, D. Stone, W. Li, A. Gurney, M. Armanini, C. Grey, A. Rosenthal, and F. J. de Sauvage, "Gli regulation by the opposing activities of fused and suppressor of fused," *Nature cell biology*, vol. 2, no. 5, pp. 310–312, 2000.

[323] Y. Koyabu, K. Nakata, K. Mizugishi, J. Aruga, and K. Mikoshiba, "Physical and functional interactions between zic and gli proteins," *Journal of Biological Chemistry*, vol. 276, no. 10, pp. 6889–6892, 2001.

[324] A. Palencia-Campos, A. Ullah, J. Nevado, R. Yıldırım, E. Unal, M. Ciorraga, P. Barruz, L. Chico, F. Piceci-Sparascio, V. Guida *et al.*, "Gli1 inactivation is associated with developmental phenotypes overlapping with ellis–van creveld syndrome," *Human molecular genetics*, vol. 26, no. 23, pp. 4556–4571, 2017.

[325] H.-W. Lo, H. Zhu, X. Cao, A. Aldrich, and F. Ali-Osman, "A novel splice variant of gli1 that promotes glioblastoma cell migration and invasion," *Cancer research*, vol. 69, no. 17, pp. 6790–6798, 2009.

[326] L. S. W. Loo, A. A. P. Soetedjo, H. H. Lau, N. H. J. Ng, S. Ghosh, L. Nguyen, V. G. Krishnan, H. Choi, X. Roca, S. Hoon *et al.*, "Bcl-xl/bcl2l1 is a critical anti-apoptotic protein that promotes the survival of differentiating pancreatic cells from human pluripotent stem cells," *Cell Death & Disease*, vol. 11, no. 5, pp. 1–18, 2020.

[327] J. M. Adams and S. Cory, "The bcl-2 protein family: arbiters of cell survival," *Science*, vol. 281, no. 5381, pp. 1322–1326, 1998.

[328] R. L. Carpenter and H.-W. Lo, "Hedgehog pathway and gli1 isoforms in human cancer," *Discovery medicine*, vol. 13, no. 69, p. 105, 2012.

[329] M. Merchant, F. F. Vajdos, M. Ultsch, H. R. Maun, U. Wendt, J. Cannon, W. Desmarais, R. A. Lazarus, A. M. de Vos, and F. J. de Sauvage, "Suppressor of fused regulates gli activity through a dual binding mechanism," *Molecular and cellular biology*, vol. 24, no. 19, pp. 8627–8641, 2004.

[330] Y. Zhang, L. Fu, X. Qi, Z. Zhang, Y. Xia, J. Jia, J. Jiang, Y. Zhao, and G. Wu, "Structural insight into the mutual recognition and regulation between suppressor of fused and gli/ci," *Nature communications*, vol. 4, no. 1, pp. 1–12, 2013.

[331] P. Kogerman, T. Grimm, L. Kogerman, D. Krause, A. B. Undén, B. Sandstedt, R. Toftgård, and P. G. Zaphiropoulos, "Mammalian suppressor-of-fused modulates nuclear–cytoplasmic shuttling of gli-1," *Nature cell biology*, vol. 1, no. 5, pp. 312–319, 1999.

[332] D. M. Stone, M. Murone, S. Luoh, W. Ye, M. P. Armanini, A. Gurney, H. Phillips, J. Brush, A. Goddard, F. J. de Sauvage *et al.*, "Characterization of the human suppressor of fused, a negative regulator of the zinc-finger transcription factor gli," *Journal of cell science*, vol. 112, no. 23, pp. 4437–4448, 1999.

[333] H. Han, J.-W. Cho, S. Lee, A. Yun, H. Kim, D. Bae, S. Yang, C. Y. Kim, M. Lee, E. Kim *et al.*, "Trrust v2: an expanded reference database of human and mouse transcriptional regulatory interactions," *Nucleic acids research*, vol. 46, no. D1, pp. D380–D386, 2018.

[334] N. K. Agarwal, C. Qu, K. Kunkulla, Y. Liu, and F. Vega, "Transcriptional regulation of serine/threonine protein kinase (akt) genes by glioma-associated oncogene homolog 1," *Journal of Biological Chemistry*, vol. 288, no. 21, pp. 15 390–15 401, 2013.

[335] M. D. Nye, L. L. Almada, M. G. Fernandez-Barrena, D. L. Marks, S. F. Elsawa, A. Vrabel, E. J. Tolosa, V. Ellenrieder, and M. E. Fernandez-Zapico, "The transcription factor gli1 interacts with smad proteins to modulate transforming growth factor β-induced gene expression in a p300/creb-binding protein-associated factor (pcaf)-dependent manner," *Journal of Biological Chemistry*, vol. 289, no. 22, pp. 15 495–15 506, 2014.

[336] H. Ruan, H. Luo, J. Wang, X. Ji, Z. Zhang, J. Wu, X. Zhang, and X. Wu, "Smoothened-independent activation of hedgehog signaling by rearranged during transfection promotes neuroblastoma cell proliferation and tumor growth," *Biochimica et Biophysica Acta (BBA)-General Subjects*, vol. 1860, no. 9, pp. 1961–1972, 2016.

[337] M. H. Shahi, A. Lorente, and J. S. Castresana, "Hedgehog signalling in medulloblastoma, glioblastoma and neuroblastoma," *Oncology reports*, vol. 19, no. 3, pp. 681–688, 2008.

[338] T. R. Gershon, A. Shirazi, L.-X. Qin, W. L. Gerald, A. M. Kenney, and N.-K. Cheung, "Enteric neural crest differentiation in ganglioneuromas implicates hedgehog signaling in peripheral neuroblastic tumor pathogenesis," *PloS one*, vol. 4, no. 10, p. e7491, 2009.

[339] T. Zhu, J. Zheng, W. Zhuo, P. Pan, M. Li, W. Zhang, H. Zhou, Y. Gao, X. Li, and Z. Liu, "Etv4 promotes breast cancer cell stemness by activating glycolysis and cxcr4-mediated sonic hedgehog signaling," *Cell Death Discovery*, vol. 7, no. 1, pp. 1–15, 2021.

[340] A. Mazzone, S. J. Gibbons, S. T. Eisenman, P. R. Strege, T. Zheng, M. D'Amato, T. Ordog, M. E. Fernandez-Zapico, Farrugia, and Gianrico, "Direct repression of anoctamin 1 (ano1) gene transcription by gli proteins," *The FASEB Journal*, vol. 33, no. 5, pp. 6632–6642, 2019.

[341] M. S. Mroz and S. J. Keely, "Epidermal growth factor chronically upregulates ca2+-dependent cl- conductance and tmem16a expression in intestinal epithelial cells," *The Journal of physiology*, vol. 590, no. 8, pp. 1907–1920, 2012.

[342] P. J. Woolf and Y. Wang, "A fuzzy logic approach to analyzing gene expression data," *Physiol. Genomics*, vol. 3, no. 1, pp. 9–15, Jun. 2000.

[343] B. Crauder, B. Evans, and A. Noell, *Functions and Change: A Modeling Approach to College Algebra*. Cengage Learning, Jun. 2013.

[344] H. Ressom, R. Reynolds, and R. S. Varghese, "Increasing the efficiency of fuzzy logic-based gene expression data analysis," *Physiol. Genomics*, vol. 13, no. 2, pp. 107–117, Apr. 2003.

[345] R. Edwards and L. Glass, "Combinatorial explosion in model gene networks," *Chaos*, vol. 10, no. 3, pp. 691–704, Sep. 2000.

[346] T. A. Al Qazlan, A. Hamdi-Cherif, and C. Kara-Mohamed, "State of the art of fuzzy methods for gene regulatory networks inference," *ScientificWorldJournal*, vol. 2015, p. 148010, Mar. 2015.

[347] H.-F. Wang and Q.-K. Chen, "General purpose computing of graphics processing unit: A survey," pp. 757–772, 2014.

[348] M. Harris and I. Gelado, "More on CUDA and graphics processing unit computing," pp. 443–456, 2017.

[349] Z. Liu and W. Ma, "Exploiting computing power on graphics processing unit," 2008.

[350] A. R. Brodtkorb, T. R. Hagen, and M. L. Sætra, "Graphics processing unit (GPU) programming strategies and trends in GPU computing," pp. 4–13, 2013.

[351] Y. Cai and S. See, *GPU Computing and Applications*. Springer, Nov. 2014.

[352] A. Amirkhani, E. I. Papageorgiou, A. Mohseni, and M. R. Mosavi, "A review of fuzzy cognitive maps in medicine: Taxonomy, methods, and applications," *Comput. Methods Programs Biomed.*, vol. 142, pp. 129–145, Apr. 2017.

[353] B. Kosko, "Fuzzy cognitive maps," pp. 65–75, 1986.

[354] I. Maraziotis, A. Dragomir, and A. Bezerianos, "Gene networks inference from expression data using a recurrent neuro-fuzzy approach," *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, vol. 2005, pp. 4834–4837, 2005.

[355] R. Manshaei, P. Sobhe Bidari, M. Aliyari Shoorehdeli, A. Feizi, T. Lohrasebi, M. A. Malboobi, M. Kyan, and J. Alirezaie, "Hybrid-controlled neurofuzzy networks analysis resulting in genetic regulatory networks reconstruction," *ISRN Bioinform*, vol. 2012, p. 419419, Nov. 2012.

[356] K. Raza, "Fuzzy logic based approaches for gene regulatory network inference," pp. 189–203, 2019.

[357] J. A. Dickerson, Z. Cox, E. S. Wurtele, and A. W. Fulmer, "Creating metabolic and regulatory network models using fuzzy cognitive maps."

[358] T. A. Al Qazlan, A. Hamdi-Cherif, and C. Kara-Mohamed, "State of the art of fuzzy methods for gene regulatory networks inference," *Scientific World Journal*, vol. 2015, p. 148010, Mar. 2015.

[359] R. Ram, M. Chetty, and T. I. Dix, "Fuzzy model for gene regulatory network."

[360] P. J. Woolf and Y. Wang, "A fuzzy logic approach to analyzing gene expression data," *Physiol. Genomics*, vol. 3, no. 1, pp. 9–15, Jun. 2000.

[361] H. Ressom, D. Wang, R. S. Varghese, and R. Reynolds, "Fuzzy logic-based gene regulatory network," May 2003.

[362] M. S. B. Sehgal, I. Gondal, L. Dooley, and R. Coppel, "AFEGRN: Adaptive fuzzy evolutionary gene regulatory network re-construction framework," 2006.

[363] ——, "Coalesce gene regulatory network reconstruction: A Cross-Platform transcriptional gene network fusion framework," 2006.

[364] A. S. B. Sehgal, I. Gondal, and L. S. Dooley, "CF-GeNe: Fuzzy framework for robust gene regulatory network inference," 2006.

[365] R. I. Hamed, "Computational modeling and dynamical analysis of genetic networks with FRBPN- algorithm," pp. 49–55, 2011.

[366] F. Wang, D. Pan, and J. Ding, "A new approach combined fuzzy clustering and bayesian networks for modeling gene regulatory networks," 2008.

[367] C. M. Poblete, F. V. Parra, J. B. Gomez, M. C. Saldias, S. S. Garrido, and H. M. Vargas, "Fuzzy logic in genetic regulatory network models," p. 363, 2009.

[368] P. Du, J. Gong, E. SyrkinWurtele, and J. A. Dickerson, "Modeling gene expression networks using fuzzy logic," pp. 1351–1359, 2005.

[369] R. I. Hamed, S. I. Ahson, and R. Parveen, "A new approach for modelling gene regulatory networks using fuzzy petri nets," *J. Integr. Bioinform.*, vol. 7, no. 1, Feb. 2010.

[370] Y. Chen, L. J. Mazlack, and L. J. Lu, "Inferring fuzzy cognitive map models for gene regulatory networks from gene expression data," 2012.

[371] L. G. Volkert and N. Malhis, "An efficient method for fuzzy identification of regulatory events in gene expression time series data."

[372] P. C. H. Ma and K. C. C. Chan, "Inferring gene regulatory networks from expression data by discovering fuzzy dependency relationships," pp. 455–465, 2008.

[373] K. Raza, "Fuzzy logic based approaches for gene regulatory network inference," *Artif. Intell. Med.*, vol. 97, pp. 189–203, Jun. 2019.

[374] W. E. Combs and J. E. Andrews, "Combinatorial rule explosion eliminated by a fuzzy rule configuration," pp. 1–11, 1998.

[375] B. A. Sokhansanj and J. P. Fitch, "URC fuzzy modeling and simulation of gene regulation."

[376] B. A. Sokhansanj, J. P. Fitch, J. N. Quong, and A. A. Quong, "Linear fuzzy gene network models obtained from microarray data by exhaustive search," *BMC Bioinformatics*, vol. 5, p. 108, Aug. 2004.

[377] M. Gormley, V. U. Akella, J. N. Quong, and A. A. Quong, "An integrated framework to model cellular phenotype as a component of biochemical networks," *Adv. Bioinformatics*, vol. 2011, p. 608295, Nov. 2011.

[378] P. Aiyetan, "A computational time complexity analyses and the multistaged hyperparallel optimization approach of the fuzzy logic mechanistic model of molecular regulation," *SUBMITTED (Under REVIEW)*.

[379] P. Aiyetan and A. Quong, "A fuzzy logic-based mechanistic model of inferred regulators of synthetic lethality in the vorinostat-resistant HCT116, colon cancer xenograft model, cell line," *IN PREPARATION*.

[380] B. A. Sokhansanj, J. P. Fitch, J. N. Quong, and A. A. Quong, "Linear fuzzy gene network models obtained from microarray data by exhaustive search," *BMC Bioinformatics*, vol. 5, p. 108, Aug. 2004.

[381] A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. D. Favera, and A. Califano, "ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context," 2006. [Online]. Available: http://dx.doi.org/10.1186/1471-2105-7-s1-s7 LB - hJaB

[382] R. E. Blahut, "Fast algorithms for signal processing," 2010.

[383] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, Jul. 2009.

[384] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," p. 107, 2008.

[385] ——, "MapReduce: a flexible data processing tool," *Commun. ACM*, vol. 53, Jan. 2010.

[386] M. Dayalan, Senior Software Developer, ANNA University, Chennai, and India, "MapReduce: Simplified data processing on large cluster," pp. 399–403, 2018.

[387] A. B. Yoo, M. A. Jette, and M. Grondona, "SLURM: Simple linux utility for resource management," pp. 44–60, 2003.

[388] M. Jette and M. Grondona, "Slurm: Simple linux utility for resource management," in *Proceedings of ClusterWorld Conference and Expo*, Jun. 2003.

[389] X.-H. Sun and Y. Chen, "Reevaluating amdahl's law in the multicore era," pp. 183–188, 2010.

[390] G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," 1967.

[391] D. E. Culler, J. P. Singh, and A. Gupta, *Parallel Computer Architecture: A Hardware/software Approach*. Gulf Professional Publishing, 1999.

[392] K. Hwang and Z. Xu, *Scalable Parallel Computing: Technology, Architecture, Programming*. McGraw-Hill Science, Engineering & Mathematics, 1998.

[393] X.-H. Sun and L. M. Ni, "Another view on parallel speedup," 1990.

[394] X. H. Sun and L. M. Ni, "Scalable problems and Memory-Bounded speedup," pp. 27–37, 1993.

[395] J. L. Gustafson, "Reevaluating amdahl's law," pp. 532–533, 1988.

[396] J. L. Gustafson, G. R. Montry, and R. E. Benner, "Development of parallel methods for a 1024-Processor hypercube," pp. 609–638, 1988.

[397] R. E. Benner, J. L. Gustafson, and R. E. Montry, "Development and analysis of scientific application programs on a 1024-processor hypercube," *Sandia Technol.*, Feb. 1988.

[398] J. Guo, H. Liu, and J. Zheng, "Synlethdb: synthetic lethality database toward discovery of selective and sensitive anticancer drug targets," *Nucleic acids research*, vol. 44, no. D1, pp. D1011–D1017, 2016.

[399] S. Das, X. Deng, K. Camphausen, and U. Shankavaram, "Discoversl: an r package for multi-omic data driven prediction of synthetic lethality in cancers," *Bioinformatics*, vol. 35, no. 4, pp. 701–702, 2019.

[400] X. Deng, S. Das, K. Valdez, K. Camphausen, and U. Shankavaram, "Sl-biodp: Multi-cancer interactive tool for prediction of synthetic lethality and response to cancer treatment," *Cancers*, vol. 11, no. 11, p. 1682, 2019.

[401] C. J. Ryan, C. J. Lord, and A. Ashworth, "Daisy: picking synthetic lethals from cancer genomes," *Cancer cell*, vol. 26, no. 3, pp. 306–308, 2014.

[402] L. Jerby-Arnon, N. Pfetzer, Y. Y. Waldman, L. McGarry, D. James, E. Shanks, B. Seashore-Ludlow, A. Weinstock, T. Geiger, P. A. Clemons *et al.*, "Predicting cancer-specific vulnerability via data-driven detection of synthetic lethality," *Cell*, vol. 158, no. 5, pp. 1199–1209, 2014.

[403] K. Tomczak, P. Czerwińska, and M. Wiznerowicz, "The cancer genome atlas (tcga): an immeasurable source of knowledge," *Contemporary oncology*, vol. 19, no. 1A, p. A68, 2015.

[404] M. D. Robinson, D. J. McCarthy, and G. K. Smyth, "edger: a bioconductor package for differential expression analysis of digital gene expression data," *Bioinformatics*, vol. 26, no. 1, pp. 139–140, 2010.

[405] A. Liberzon, A. Subramanian, R. Pinchback, H. Thorvaldsdóttir, P. Tamayo, and J. P. Mesirov, "Molecular signatures database (msigdb) 3.0," *Bioinformatics*, vol. 27, no. 12, pp. 1739–1740, 2011.

Curriculum Vitae

Paul Aiyetan graduated as a Chevron/NNPC National Merit Scholar from the College of Medicine, University of Ibadan where he obtained his MD degree. He completed his intern year at the University College Hospital. He was thereafter, a Resident Medical Officer at the Niger Hospital. He obtained the Johns Hopkins University Zanvyl-Krieger School of Arts and Sciences and Whiting School of Engineering jointly offered MS in Bioinformatics. And, he completed a Postdoctoral Fellowship in Pathology at the Johns Hopkins University School of Medicine. Dr. Aiyetan was a member of the National Institute of Health (NIH) National Cancer Institute (NCI) funded Clinical Proteomics Tumor Analysis Consortium (CPTAC) and, the National Heart, Lung and Blood Institute (NHLBI) funded Programs of Excellence in Glycosciences (PEG). He was a recipient of Johns Hopkins Pathology Young Investigators Excellence Award in 2012 and a recipient of the inaugural NIH funded Big Data to Knowledge fellowship at the University of Minnesota and the Mayo Clinic in 2015. Dr. Aiyetan currently work at the Frederick National Laboratory for Cancer Research. His research interest is at the intersection of Medicine, Translational Research, Computation, Quantitative and Data science – leveraging computational and quantitative approaches to elaborate the etiopathogenesis of disease processes for better *precise* diagnostics and *precise* therapeutics identification and design, including for primary, secondary and tertiary preventive measures. Dr. Aiyetan has over 15 peer-reviewed publications (PubMed) and has given over 15 presentations (including talks and posters). He has peer-reviewed for BMC Bioinformatics, Journal of Proteome Research, and the 'Genes and Immunity' scientific journals.