Network Architecture for Prediction of Emergence in Complex Biological Systems

By

Gourab Ghosh Roy
ORCID: 0000-0001-9420-5653

A thesis submitted to
the University of Birmingham and the University of Melbourne
for the degree of
DOCTOR OF PHILOSOPHY

School of Computer Science
College of Engineering and Physical Sciences
The University of Birmingham

School of Computing and Information Systems
Faculty of Engineering and Information Technology
The University of Melbourne

March 2022

ABSTRACT

Emergence of properties at the system level, where these properties are not observed at the individual entity level, is an important feature of complex systems. Biological system emergent properties have critical roles in the functioning of organisms and the disruptions to normal functioning, and are relevant to the treatment of diseases like cancer. Complex biological systems can be modeled by abstractions in the form of molecular networks like gene regulatory networks (GRNs) and signaling networks with nodes representing molecules like genes and edges representing molecular interactions. The thesis aims at exploring the use of the architecture of these networks to predict emergence of system properties.

First, to better infer the network architecture with aspects that can be useful in predicting emergence, we propose a novel algorithm Polynomial Lasso Bagging or PoLoBag for signed GRN inference from gene expression data. The GRN edge signs represent the nature of the regulatory relationships, activating or inhibitory. Our algorithm gives more accurate signed inference compared to state-of-the-art algorithms, and overcomes their weaknesses by also inferring edge directions and cycles. We also show how combining signed GRN architecture with dynamical information in our proposed dynamical K-core method predicts emergent states of the gene regulatory system effectively.

Second, we investigate the existence of the bow-tie architectural organization in the GRNs of species of widely varying complexity. Prior work has shown the existence of this bow-tie feature in the GRNs of only some eukaryotes. Our investigation covers GRNs of

prokaryotes to unicellular and multicellular eukaryotes. We find that the observed bow-tie architecture is a characteristic feature of GRNs. Based on differences that we observe in the bow-tie architectures across species, we predict a trend in the emergence of the dynamical gene regulatory system property of controllability with varying species complexity.

Third, from input genotype data we predict an emergent phenotype at the organism level – the cancer-specific survival risk. We propose a novel Mutated Pathway Visible Neural Network or MPVNN, designed using prior knowledge of signaling network architecture and additional mutation data-based edge randomization. This randomization models how known signaling network architecture changes for a particular cancer type, which is not modeled by state-of-the-art visible neural networks. We suggest that MPVNN performs cancer-specific risk prediction better than other similar sized NN and non-NN survival analysis methods, while also providing reliable interpretations of the predictions.

These three research contributions taken together make significant advances towards our goal of using molecular network architecture for better prediction of emergence, which can inform treatment decisions and lead to novel therapeutic approaches and is of value to computational biologists and clinicians.

DECLARATION OF AUTHORSHIP

I, GOURAB GHOSH ROY, declare that this thesis titled, 'UNDERSTANDING EMERGENCE IN COMPLEX BIOLOGICAL SYSTEMS USING NETWORK ARCHITECTURE' and the work presented in it are my own. I confirm that:

- The thesis comprises only my original work towards the DOCTOR OF PHILOSOPHY except where indicated in the preface;
- due acknowledgement has been made in the text to all other material used; and
- the thesis is fewer than the maximum word limit in length, exclusive of tables, maps, bibliographies and appendices as approved by the Research Higher Degrees Committee.

Signed: Gourab Ghosh Roy

Date: 01/03/2022

PREFACE

- The work towards the thesis was supported by a Priestley Scholarship for joint study at the University of Birmingham and the University of Melbourne;
- The work towards the thesis was under the guidance of PhD supervisors Dr. Shan He from the University of Birmingham and Dr. Nicholas Geard and Prof. Karin Verspoor from the University of Melbourne;
- The PhD supervisors were co-authors in the publications included in the thesis;
- The published papers went through peer review and publication editing process;
- The publication status of the chapters detailing the work towards the thesis is given below;
  - Chapter 3 includes work published by Bioinformatics on Jul 22, 2020 and some unpublished material not submitted for publication.
  - Chapter 4 includes work published by Journal of The Royal Society Interface on June 09, 2021.
  - Chapter 5 includes work submitted for publication to Bioinformatics on Feb 02, 2022 and also available at arXiv.

PUBLICATIONS

1. **Roy GG**, Geard N, Verspoor K, He S. PoLoBag: Polynomial Lasso Bagging for signed gene regulatory network inference from expression data. Bioinformatics. 2020 Jul 22.

2. **Roy GG**, He S, Geard N, Verspoor K. Bow-tie architecture of gene regulatory networks in species of varying complexity. Journal of The Royal Society Interface. 2021 June 09.

   This work is also presented as a poster at NetBio COSI, Intelligent Systems For Molecular Biology/European Conference On Computational Biology 2021. 2021 July 25-30.

3. **Roy GG**, Geard N, Verspoor K, He S. MPVNN: Mutated Pathway Visible Neural Network architecture for interpretable prediction of cancer-specific survival risk. Bioinformatics. Submitted 2022 Feb 02.

   This work is also available at arXiv.

DEDICATION

Dedicated to my wonderful and immensely supportive parents Mr. Goutam Ghose Roy and Mrs. Seema Ghose Roy. Words can not express how grateful I am. This is as much their doctoral thesis as it is mine.

ACKNOWLEDGMENTS

I wish to acknowledge the valuable guidance of my PhD supervisors Dr. Shan He, Dr. Nicholas Geard and Prof. Karin Verspoor all through my PhD work. They have constantly motivated me to try to answer difficult research questions, supported me through ups and downs in this journey, and their own work ethics and values have greatly inspired me in this PhD and my research career going forward.

I am very grateful to the University of Birmingham and the University of Melbourne for giving me this joint PhD opportunity through the funding support of the Priestley scholarship. I think the collaboration between the universities has provided a very unique and fruitful research experience.

I also want to sincerely thank Prof. Ata Kaban and Dr. Per Kristian Lehre from the University of Birmingham and Dr. Rajkumar Buyya from the University of Melbourne for supervising my progress at regular intervals and giving me valuable suggestions through my PhD. I would like to express my gratitude towards the great staff at both universities for helping me with all the administrative queries.

I would also like to express my gratitude towards Prof. Asa Ben-Hur and Dr. Ian Overton for agreeing to be the examiners of my thesis and viva.

My special thanks to my friends and colleagues at both Universities, particularly in the NAD (complex Network Architecture and Dynamics) group and Office No. 144 at the University of Birmingham, where I have spent the most time during my PhD.

In the case of all things which have several parts
and in which the totality is not, as it were, a mere heap,
but the whole is something besides the parts

Aristotle, "Metaphysics"

Contents

List of Figures

List of Tables

List of Algorithms

LIST OF DEFINITIONS FOR ALGORITHMS

1. Algorithm 1

   - $\boldsymbol{w}$ – Signed edge weight column vector,
   - $\boldsymbol{w}^t$ – Signed edge weight column vector for a target gene t,
   - $\boldsymbol{w}_M^t$ – Weight magnitude column vector for a target gene t,
   - $\boldsymbol{w}_S^t$ – Weight sign column vector for a target gene t,
   - $\boldsymbol{s}_w^t$ – Column vector to store number of times each regulator expression profile is used in features across all bootstrap samples for target gene t,
   - n_R^t – Number of potential regulators for target gene t,
   - n_d^k – d user defined values to control the number of polynomial features for every $k \in \{1, \cdots, d\}$, where d is the defined polynomial degree of the algorithm,
   - $\boldsymbol{n}_F^t$ – Number of polynomial features for every $k \in \{1, \cdots, d\}$ in a bootstrap sample for target gene t,
   - n_B – User defined number of bootstrap samples,
   - n_M – User defined bootstrap sample size, after multiplication with total number of measurement conditions m,
   - $\boldsymbol{id}^{tb}$ – Stores the individual input feature indices and the category c for every feature selected in a bootstrap sample b for target gene t.

2. Algorithm 2

   - v – Vertex,
   - $DFS_G(v)$ – The set of vertices from a depth-first search starting at v in network G,
   - $DFS_{G^T}(v)$ – The set of vertices from a depth-first search starting at v in network G^T which is obtained by reversing the direction of every edge in G.

3. Algorithm 3

   - W – Neural network connection matrix denoting connections between N neurons in the input layer and N neurons in the intermediate layer,
   - W_{ab} – One element of W denoting whether a connection exists between input layer expression neuron for gene a and intermediate layer perturbation neuron for gene b,
   - $M_{k,n}$ – Mutation for gene Gn in sample k, 1 is non-silent mutation, 0 is wild type,
   - $frac_p$ – Fraction of total N genes in pathway p,
   - thr_p – Mutation threshold for pathway p,
   - I_N – Identity matrix of size N,
   - E_{new} – Stores replacing edges,
   - $random()$ - Random number in $[0, 1)$.

Chapter One

Introduction

1.1 Background

Our world is filled with complex systems, where simple entities interact to produce higher-level behaviors. The ubiquitous phenomenon of emergence is considered to be a characteristic feature of complex systems (Boschetti et al., 2005). A system property can be defined as emergent if it is observed at the level of the system and not at the level of lower-level entities (Baas and Emmeche, 1997). One major area of analysis of emergence is in the field of biological systems. Study of biological systems helps us comprehend how living organisms work. These systems exist at different levels of biological hierarchy, for example genes, cells, tissues and organisms. Prediction of emergence can tell us the fate of a cell or ultimately the overall organism. One of the main objectives of systems biology is to understand emergence (Tavassoly, Goldfarb, and Iyengar, 2018). Despite significant advances in the study of functionalities related to many biological systems, there are still considerable gaps in fully and reliably understanding and predicting how system-level behavior emerges. The ability to better predict emergent properties associated with the functioning of these critical biological systems can improve our knowledge of these systems and guide development of therapeutic

approaches leveraging such properties.

One primary tool in the very challenging task of understanding emergence is understanding information processing (C. R. Shalizi, K. L. Shalizi, and Crutchfield, 2002). Emergence of biological system behavior could be understood by analyzing how information is processed within the system (Nurse, 2008). This term processing here encompasses gathering, modifying, transmitting, using or storing the information. To understand how information is processed in a complex system, abstractions of the system can be used, abstractions which can accurately model the system and are naturally designed for analysis of information processing. An example of such an abstraction is complex networks, which are defined as graphs consisting of individual nodes connected by edges, having topological features that are not completely regular nor completely random. Many real-world systems in major fields like social sciences, electrical engineering, and in this context biology can be modeled using complex networks (J. Kim and Wilhelm, 2008; Barabási, Gulbahce, and Loscalzo, 2011).

Biological information processing is accomplished by the molecular networks representing interactions between molecules like genes, proteins, metabolites, etc. Gene regulatory networks (GRNs) (Karlebach and Shamir, 2008) are networks that represent interactions of gene regulation between regulators like transcription factors and their target genes. GRNs are considered to be the downstream parts of signaling networks, which are molecular networks representing interactions involved in cellular signaling. Signaling networks are made up of individual signaling pathways (Weng, Bhalla, and Iyengar, 1999). Emergence of cellular behavior being controlled by information processing in these molecular networks is depicted in Figure 1.1.

The network architecture refers to the arrangement of nodes in a network. The architecture captures all the structural information in the network, like existence of edges between

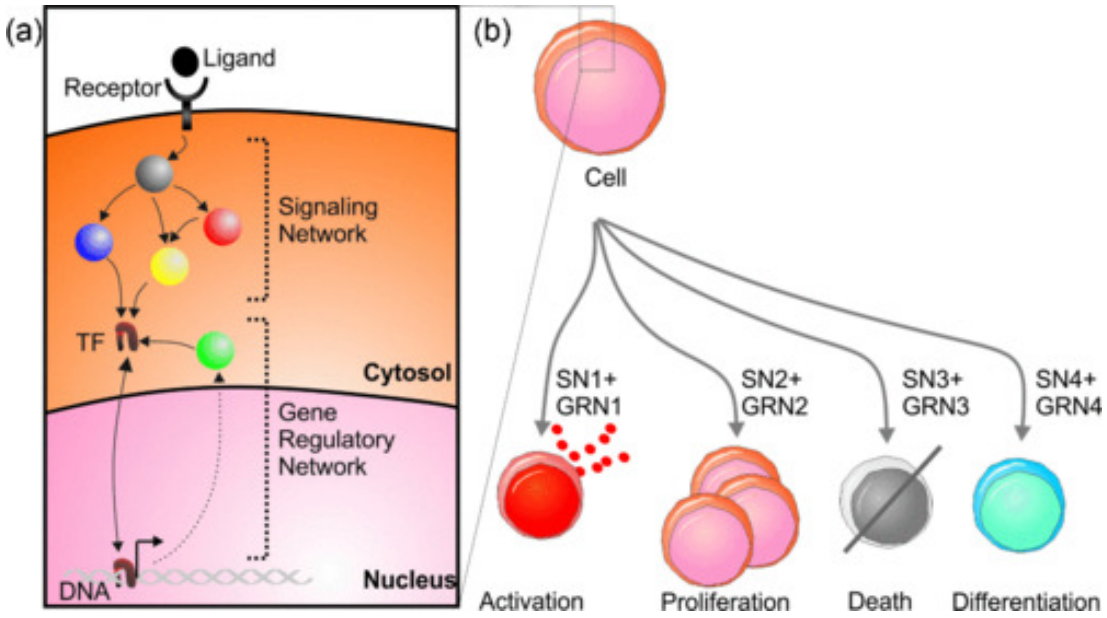


Figure 1.1: Information processing in molecular networks for cellular decision-making. (a) Information processing through signaling and gene regulatory networks, (b) Cellular decision-making is based on information processing in these networks, where different combinations of activated signaling and gene regulatory networks lead to different behaviors. Reprinted figure with permission from Ferreira, Nakaya, and Fontoura Costa, 2018. Copyright (2018) by the American Physical Society.

nodes, and directions and signs of those edges. Biological systems are dynamical systems, therefore their state changes over time under different conditions. Network dynamics refers to how a network's state changes. Information is processed through the change of network states, and this change or network dynamics is governed by how the molecules in the network are arranged, that is the network architecture. How the molecules interact with each other is also important, which is denoted by dynamical information like interaction functions and associated parameters. The architecture and the dynamical information need to be considered together in the model to have a complete picture of emergence (Tavassoly, Goldfarb, and Iyengar, 2018). This important relationship between network architecture and dynamics has been explored (Tyson, K. C. Chen, and Novak, 2003; Alon, 2007; J.-R. Kim, Yoon, and K.-H. Cho, 2008). However, this is done usually at small network sizes, since there are too many unknowns to model for large networks.

One way of obtaining architectures of GRNs is using curated databases (Santos-Zavaleta et al., 2019; Z.-P. Liu et al., 2015). However the architecture information is not always complete. Network edges might be missing, and additionally information like edge signs representing the activating or inhibitory nature of regulatory relationships is missing. Another way of deriving GRN architectures is GRN inference, as with the availability of high quality gene expression data from experiments, the network architecture can be accurately reverse engineered from the data (Gardner and Faith, 2005). An important concept in network reverse engineering is identifiability denoting whether a network can be uniquely determined from the available data (Zak et al., 2003). In GRN inference, there are usually lesser measurement samples than the number of genes in the network, and also there is noise in gene expression data. For an inference algorithm, these limitations in the data must be balanced with the inferred GRN complexity, where important architectural aspects should be considered to accurately represent the regulatory system. However, many popular GRN inference algorithms (Margolin et al., 2006; Huynh-Thu, Irrthum, et al., 2010) do not

produce edge signs. Algorithms for signed inference suffer from some key limitations such as not producing edge directions (Khosravi et al., 2015) or network cycles (J. Yu et al., 2004), both of which along with edge signs are aspects of the architecture which can be useful in prediction of emergence.

Emergent properties are associated with network architectural features denoting how the nodes fit in an organization within a network. Example of such an architectural feature is the bow-tie architecture (R. Yang, Zhuhadar, and Nasraoui, 2011). This feature is associated with emergent properties like controllability, and understanding how these properties emerge would be helpful in understanding how diseases develop and therapeutic methods can be designed (Kitano, 2004a). Prior work has shown the existence of a bow-tie architecture in GRNs of some eukaryotes (Rodriguez-Caso, Corominas-Murtra, and Solé, 2009; S. Luo et al., 2018). However, a more complete investigation of the existence of the bow-tie architecture across GRNs in species of a wide range of biological complexity has not yet been performed. In such an investigation, quantification of the characteristics of the bow-tie architecture with varying species complexity can predict a trend of change in emergence of the important gene regulatory system property of controllability.

Improved prediction of emergence involves not just predicting general trends but predicting more specific values of emergent properties. One practical application of predicting an emergent phenotype is to predict cancer-specific survival risk of a patient from genotypic measures. Cancer survival risk prediction using gene expression has been studied as a machine learning task. Neural networks (NNs) are a well-known category of machine learning methods, and they have been used for prediction of cancer risk (Katzman et al., 2018; Z. Huang et al., 2019). However, a major challenge of standard fully connected NNs is that they are used as black boxes, with lack of interpretability or understanding of their internal operations in biological terms. This lack of interpretability leads to lack of user trust in the model. To increase the model interpretability, there are visible NNs or VNNs where neurons

represent biological entities like genes, proteins, pathways, cell subsystems, etc. and connections between the neurons represent biological relationships. Biological pathway knowledge is used in the design of VNNs (Elmarakeby et al., 2021; Hilten et al., 2020; J. Ma et al., 2018; Fortelny and Bock, 2020). However, the state-of-the-art VNNs do not model how known biological pathway architecture can change for particular types of cancer. Flow of information through pathways can be disrupted in a particular way for a particular cancer. Cancer survival risk is also affected by gene mutations, for instance BRCA1/2 mutations in breast and ovarian cancer (Kurian, Sigal, and Plevritis, 2010). Gene mutation data can be used to simulate the disruption of information flow. Since the design of VNNs is based on biological relationships, consideration of this disruption in the VNN model can be useful in the prediction of cancer-specific survival risk.

1.2 Research Questions

Based on the identified research gaps in the background work, our aim is to better predict emergence in biological systems using molecular network architecture. We work towards that aim by addressing three key research questions outlined below.

1. **How to better infer GRN architecture from data?**

   The first research question is how to more accurately infer the signed GRN architecture from gene expression data. A general form of expression data is considered, that is without any prior time course or gene knock-out assumptions or availability of reference wild-type measurements. The question also covers how to simultaneously infer other architectural aspects like edge directions and cycles which along with edge signs can play important roles in predicting emergence in the gene regulatory system.

2. **Can trends in emergence be predicted using a characteristic feature of GRN**

architectures?

The second research question is if there exists an architectural feature – the bow-tie architecture in the GRNs of species of widely varying biological complexity. The next part of the question is if there are quantitative traits of the bow-tie architecture that can be analyzed for understanding differences in the emergence of the dynamical gene regulatory system property of controllability with species biological complexity.

3. **How to better predict organism-level emergence using signaling network architecture?**

   The third research question is how an organism-level emergent phenotype of the cancer-specific survival risk can be effectively predicted from genotype using knowledge of signaling network architecture and modeling of how the architecture can change for particular cancer types. This also covers if the predictions can be reliably interpreted to provide insights which correspond to the actual emergence of risk.

1.3 Research Contributions

The contributions of the thesis are summarized in this section. We answer the three proposed research questions as follows.

First, we propose an algorithm Polynomial Lasso Bagging or PoLoBag for signed gene regulatory network (GRN) inference from a general form of gene expression data without any prior time course or gene knock-out assumptions or availability of reference wild-type measurements. We demonstrate that our algorithm consistently performs more accurate signed inference compared to state-of-the-art algorithms on simulated and real-world expression datasets. Our algorithm also overcomes the key shortcomings of other algorithms as it infers signed networks with both edge directions and network cycles. Additionally, we

combine signed GRN architecture and dynamical information to propose a dynamical K-core decomposition method for finding top regulators in the GRN and suggest that the top regulators identified by our method predict emergent states of the gene regulatory system under different measurement conditions better than those identified from the K-core decomposition method.

Second, we find the existence of a bow-tie architectural feature in the GRNs of species of widely varying complexity from prokaryotes to unicellular and multicellular eukaryotes including human. We show that this is a characteristic feature of these GRNs, which can not be explained just by chance. Based on the observed quantitative differences of the GRN bow-tie architectures, we hypothesize how the dynamical gene regulatory system property of controllability has emerged differently with species complexity.

Third, we propose a novel Mutated Pathway Visible Neural Network or MPVNN for predicting emergent phenotypic property at the organism level – cancer-specific survival risk from genotype gene expression data. The proposed neural network is designed using prior knowledge of signaling network architecture and mutation data-based edge randomization simulating how the known signaling network architecture changes for a particular cancer type. We show that our MPVNN can give better overall cancer-specific survival risk prediction mean performance than similar sized NN and other standard non-NN methods. Importantly, this visible neural network can be interpreted to provide insights about sets of genes linked by flow of signal that are important in cancer-specific risk prediction, and from literature validation we argue that these insights are reliable corroborating with risk emergence.

From our work, we are able to accurately infer signed GRN architecture that is found to be useful in predicting emergence in the gene regulatory system. Our work on the bow-tie architecture in GRNs of species of widely varying biological complexity enables us to predict a trend in the emergence of the dynamical gene regulatory system property of controllabil-

ity with varying species complexity. Lastly, our work on the visible neural network using signaling network architecture and modeling how the known architecture changes for particular cancer types allows us to effectively predict the emergent phenotype of cancer-specific survival risk in an interpretable manner. Therefore, putting forward these three contributions, we make significant advances towards our central thesis objective of using network architecture for predicting emergence in complex biological systems.

1.4 Thesis Outline

The outline of the overall thesis is given in this section. In Chapter 1 we have introduced the research gaps in the background work, the proposed research questions and our contributions. For the remaining chapters, Chapter 2 presents a high-level literature review, followed by details of three individual studies. In Chapter 3, we first present our signed GRN inference algorithm PoLoBag, and then describe our dynamical K-core method combining signed GRN architecture and dynamical information for emergent state prediction. Chapter 4 describes our work on investigation of the bow-tie architecture in GRNs of species of widely varying complexity to predict a trend in the emergence of the dynamical gene regulatory system property of controllability. The work on MPVNN neural network for predicting the emergent phenotype of cancer-specific survival risk is presented in Chapter 5. Chapter 6 summarizes what we learn from the overall work done in the thesis and discusses some examples of future work.

Chapter Two

Literature Review

In this chapter we present a literature review of the prior work and concepts relevant to our central objective of using network architecture for predicting emergence in complex biological systems. First we review the definitions of emergent properties and biological networks which form the basis of our work. We present reviews of network architecture knowledge, architecture inference from data and network architectural features. Next we present some popular machine learning methods used for the task of prediction. Finally we review prediction of emergence using network architecture. Here we refer the reader to the sections in respective chapters for very specific details of prior work and concepts, as those details are useful in illustrating how each of the individual studies is motivated by and still very novel relative to the relevant prior work.

2.1 Emergent Properties

Much effort has been devoted to formally defining the concept of emergence (Kubí, 2003). In this thesis, we use the simple and elegant definition where a system property is defined as an emergent property if it is displayed at the level of the system and not displayed by lower-level

entities within the system (Baas and Emmeche, 1997). More formally, let the overall system be modeled by a network N^2, made up of individual nodes ${N_i}^1$. Here superscripts 1 and 2 represent the two levels of this simple hierarchy. We consider an observational mechanism Obs^j where $j \in \{1, 2\}$. Now a property P is defined to be an emergent property iff

$$P \in Obs^2(N^2), \quad P \notin Obs^2({N_i}^1) \text{ for all } i. \tag{2.1}$$

Here we refer to emergent behavior under one particular condition as an emergent state, and an emergent property usually encapsulates behavior under multiple conditions.

A biological system emergent property is associated with the hierarchical level of the system. Here we consider emergent properties at the level of the gene regulatory system and at the level of the entire organism. The gene regulatory system emergent property of interest in this thesis is controllability. A non-linear dynamical system is defined to be controllable when there is a control path from an undesired attractor state to a desired attractor state under finite perturbations, where attractor states are stable equilibrium states in the phase space (L.-Z. Wang et al., 2016). Controllability is universally present in the gene regulatory system and can be utilized in therapy since cancer cells are trapped in abnormal attractor states (S. Huang, Ernberg, and S. Kauffman, 2009).

One particular property which is very clinically significant and emergent at the organism level is the cancer survival risk of a patient. This property, indicative of the survival time of a patient (J. Liu et al., 2018), can be expressed as a risk score (Katzman et al., 2018; Z. Huang et al., 2019). Being able to effectively predict this phenotype helps in making informed decisions about patient treatment. For cancer-specific survival, an event refers to death specifically from the diagnosed cancer type (J. Liu et al., 2018). We further discuss the property of cancer-specific survival risk in Section 5.2.1. This emergent property can inform patient treatment by guiding the suitability of a treatment method or by assessing the effectiveness of an applied method.

2.2 Biological Networks

One common representation used to model biological systems is that of biological networks. A network here is defined as $G = (V, E)$, where V denotes the set of nodes representing biological entities, and E denotes a set of edges or connections representing biological relationships between those entities. Edges can have attributes like direction, sign, weight, etc. This is illustrated in Figure 2.1. For a network, the architecture refers to the arrangement of nodes and the dynamics refers to how the states change over time under varying conditions.

In this thesis we are studying two particular molecular networks – gene regulatory networks (GRNs) and signaling networks. GRNs represent interactions between regulators like transcription factors and target genes. Through binding, a transcription factor controls the rate of transcription of its target gene, either activating or repressing the transcription. The expression of genes are controlled by these GRNs. For example, the transcription factor MYC upregulating CDK4 (Hermeking et al., 2000) is represented by the GRN edge MYC→CDK4. The process of transcription is also affected by regulators like prokaryotic sigma factors and chromatin remodeling factors. Apart from transcriptional relationships, GRN edges can also represent post-transcriptional relationships between regulators like bacterial small RNAs or microRNAs and their target genes. Signaling networks are made up of individual signaling pathways representing molecular interactions involved in cellular signaling. Analysis of these networks is useful in understanding normal functioning and disease development in organisms.

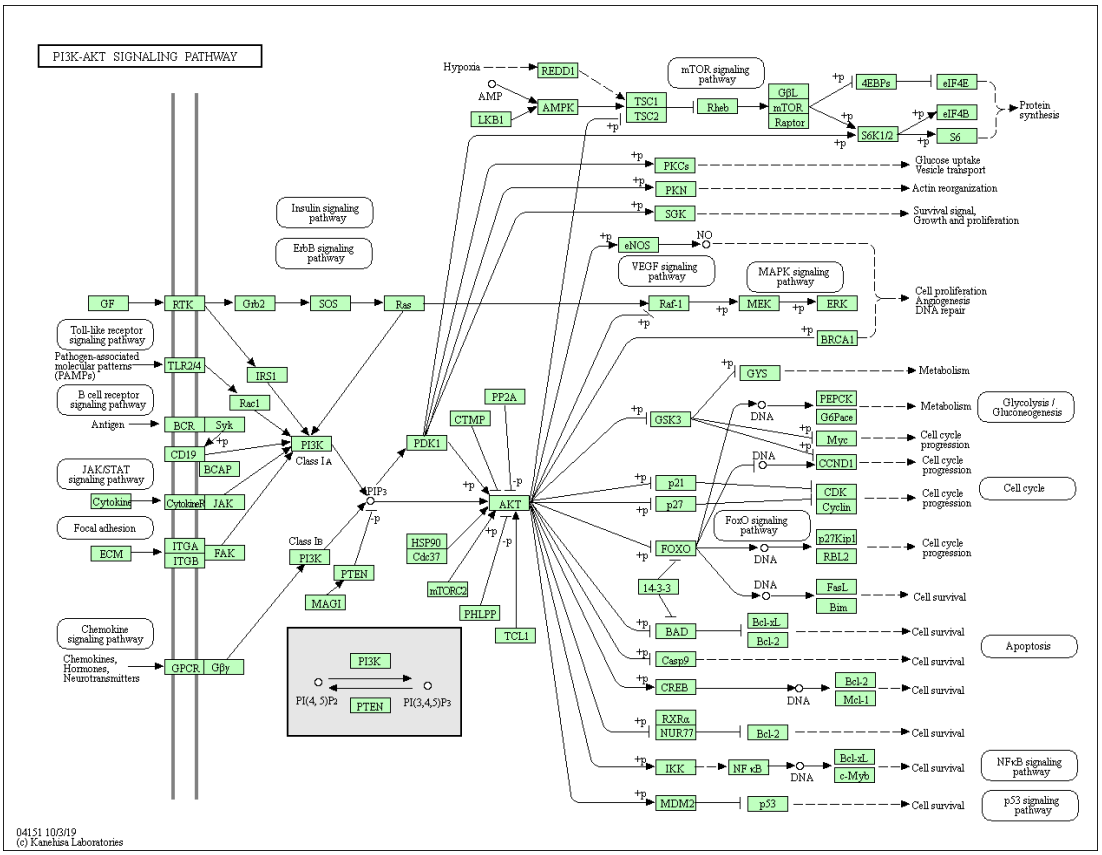


Figure 2.1: The PI3K-Akt signaling pathway (Kanehisa and Goto, 2000).

2.3 Network Architecture Knowledge

Owing to the substantial amount of research that has been done in the study of molecular networks, it has been possible to construct databases for GRNs and signaling networks. For GRNs, there are different databases which focus on individual species or groups of species. Examples of such GRN databases include RegulonDB (Santos-Zavaleta et al., 2019), YTRP (T.-H. Yang et al., 2014), AtRegNet (Yilmaz et al., 2010), DroID (Murali et al., 2011), etc. for non-human species. Human GRNs can be obtained from data sources like RegNetwork (Z.-P. Liu et al., 2015), TRRUST (H. Han et al., 2018), ORTI (Vafaee et al., 2016), etc. Further details of databases are given in Section 4.2.1. These contain regulator and target gene interactions collected using different methodologies, where some interactions are experimentally verified and some are predicted from computational techniques. Examples of regulators are transcription factors, microRNAs, etc. The interactions are ranked based on the reliability of the associated evidence. Different data sources use their own set of criteria for defining the interaction ranks, and in some cases the information is not available. Completeness in terms of coverage of genes is also a major issue. The architecture of signaling networks can be obtained from the KEGG Pathway database (Kanehisa, Furumichi, et al., 2021), the Reactome Pathway database (Griss et al., 2020), etc. These discussed databases can be used as data sources for subsequent architecture analysis of GRNs and signaling networks.

2.4 Network Architecture Inference

In the architectures of GRNs that are obtained from databases, some vital information can be missing, for example in the form of missing edges or unavailability of edge signs. GRN inference is the reverse-engineering of the GRN from gene expression data (Gardner and

Faith, 2005). Gene expression data, collected from real biological experiments or simulated from GRN models, can be of several types (Schaffter, Marbach, and Floreano, 2011). Expression data can be steady-state or time course. It can be categorized as single or double gene knockout or knockdown, referring to genes being inoperative fully or partially, or multifactorial data where basal levels of all genes are perturbed. The steady-state multifactorial data is a more commonly available form of gene expression data.

Several categories of GRN inference methods exist, and each category, based on its strengths and weaknesses, is useful in different settings. One type of methods are based on the Boolean network model (S. A. Kauffman, 1969). These simple Boolean network models represent genes as Boolean variables whose values are updated in time steps. Another category of methods are based on Bayesian networks, which are used to represent the probabilistic relationships between variables or genes. An example is Bayesian Network Inference with Java Objects (Banjo) (J. Yu et al., 2004). As these Bayesian networks are acyclic graphs, static version of the Banjo cannot have cycles in the inferred GRN. Information theoretic inference methods aim to uncover statistical dependencies between the expression profiles of genes. One such algorithm using the measure of mutual information (MI) is the Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNe) (Margolin et al., 2006). Information theoretic algorithms do not infer network edge directions.

A popular category of inference methods are the ones based on regression. In contrast to simpler Boolean models, the continuous expression values of a target gene are considered a function of the continuous expression values of the regulator genes. For classification-based methods, the target gene expression can be categorized into up or down states (Middendorf et al., 2004). An example of a regression-based inference method is Gene Network Inference with Ensemble of trees (GENIE3) (Huynh-Thu, Irrthum, et al., 2010) which uses tree-based ensemble approaches. Another type of methods use biologically more realistic ordinary differential equations (ODE) models. These complex models relate the rate of change of

expression of a target gene to its own expression and the expression of regulator genes, and can be simplified to regression models for steady-state expression data. An example is Inferelator (Bonneau et al., 2006).

Many popular inference algorithms do not give edge signs denoting the activating or inhibitory nature of the regulatory relationships. We are interested in signed GRN inference algorithms which can infer from a general form of expression data, that is without any time course or gene knockout assumptions or availability of reference measurements. Example of such an algorithm is Banjo. Static Banjo produces edge signs, but cannot have cycles in the inferred GRN. A signed inference algorithm from the category of information theoretic methods is Signing of Regulatory Networks (SIREN) (Khosravi et al., 2015). But this algorithm does not give edge directions. Some signed inference algorithms apply the Least absolute shrinkage and selection operator (Lasso) (Tibshirani, 1996). Lasso is a variable selector which selects input variables or features that can best explain the output variable. However, as pointed out in Zou and T. Hastie, 2005, in Lasso on its own the variable selection is firstly constrained by the sample size and secondly is not accurate when there are highly correlated input variables as is the case with GRNs where many regulators in synergy can regulate a target gene. We present further details of signed GRN inference algorithms in Section 3.1. From the literature review it is observed that state-of-the-art inference algorithms suffer from some major limitations which need to be addressed:

- not inferring edge signs,
- not inferring edge directions or network cycles simultaneously when inferring edge signs,
- Lasso for inferring edge signs not being accurate on its own.

2.5 Network Architectural Features

Different nodes in the GRN architecture play different roles in controlling the behavior of the network. An approach of simplifying the analysis of the GRN architecture is to find out which of the nodes play more critical roles. However, identification of important nodes is not an easy task, and the GRN of a human has been compared with a tangled hairball (Narang et al., 2015). Different measures of importance can be used to rank nodes in the architecture, like those based on node degree denoting its number of connections or node centrality denoting its relative position compared to other nodes in the network (Borgatti, 2005).

A well-known node ranking method is the K-core method. The K-core of a network is a maximum subnetwork where the degree of every node is greater than or equal to K (Seidman, 1983). Figure 2.2 shows the cores of a network. The K-core network decomposition method (Batagelj and Zaversnik, 2003) can identify a hierarchical organization of nodes as done for GRNs of bacteria (Malkoç, Balcan, and Erzan, 2010), yeast *Saccharomyces cerevisiae* (Balcan et al., 2007), human (Narang et al., 2015). Section 3.5.1 elaborates further on this. The inner cores identified by K-core decomposition points to which regulator nodes are most important.

An architectural feature denotes how the nodes fit into a particular organization in the network. In the core-periphery architectural feature, some nodes form a central densely connected core (Csermely et al., 2013). The bow-tie architecture is a particular case of core-periphery architecture. A bow-tie architecture in a directed network is defined with a strongly connected CORE layer which lies between the IN layer and the OUT layer (R. Yang, Zhuhadar, and Nasraoui, 2011). A strong component is a subnetwork where every node is connected to every other node, and the largest of these components is defined to be the CORE layer (Broder et al., 2011). An example of a network bow-tie architecture is

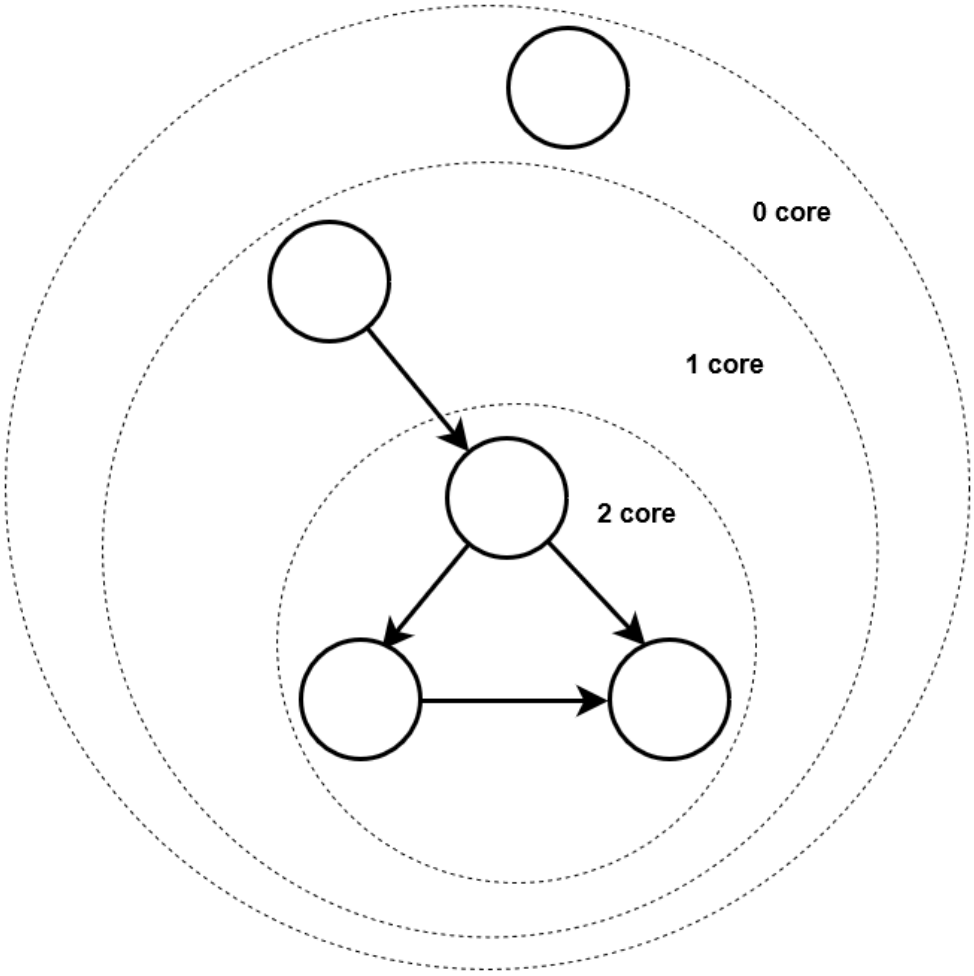


Figure 2.2: Network K-core – 0, 1 and 2 core of a network. The circles represent nodes and the arrows represent edges. The node degree is in-degree + out-degree. The different cores are denoted by dashed circles.

given in Figure 2.3. Further details of the bow-tie architectural feature is given in Sections 4.1 and 4.2.3. A bow-tie architecture has been previously observed in the GRNs of two eukaryotic species – Yeast (*Saccharomyces cerevisiae*) (Rodriguez-Caso, Corominas-Murtra, and Solé, 2009) and Arabidopsis (*Arabidopsis thaliana*) (S. Luo et al., 2018). However no study has yet investigated the existence of the bow-tie architecture and the quantification of its characteristics across GRNs in species of a wide range of biological complexity from

bacteria to human.

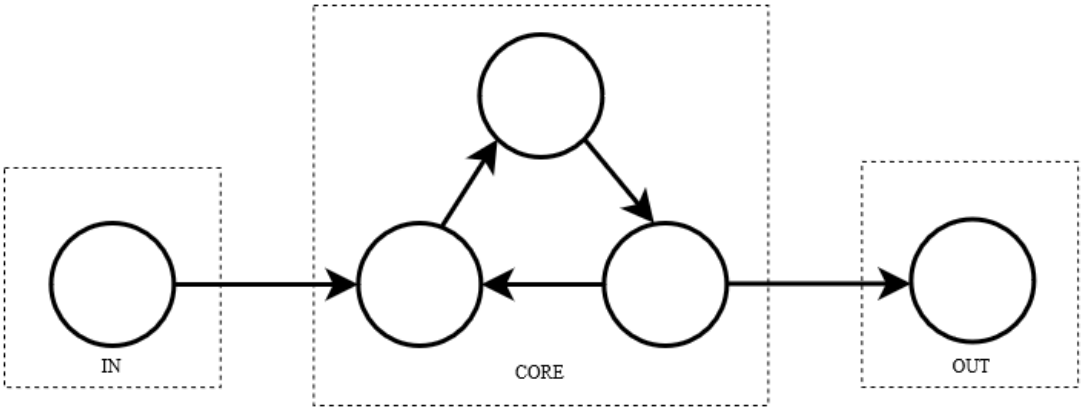


Figure 2.3: An example of a bow-tie architecture with the largest strong component (LSC) CORE layer. The circles represent nodes and the arrows represent edges. The bow-tie layers CORE, IN and OUT are denoted by dashed boxes.

2.6 Prediction

There are different categories of machine learning methods that are used for the task of prediction. One such popular method is support vector machine (Cortes and V. Vapnik, 1995). In a binary classification setting, support vector machine (SVM) works by constructing a hyperplane that can separate the data into two classes with maximum separation between the classes. SVM is a linear classifier, however for cases when the data is not linearly separable, the kernel approach (Boser, Guyon, and V. N. Vapnik, 1992) constructs the hyperplane in a transformed space. An extension of the SVM method referred to as support vector regression is used in a regression setting (Drucker et al., 1996).

Another category of machine learning methods for prediction is that of neural networks (McCulloch and Pitts, 1943). The basic unit of a neural network is a neuron, which

is connected to other such neurons. These neurons are arranged in layers. The output of a neuron in layer l is given as $x_{out} = f(w * x_{in} + b)$, where x_{in} represents the values of the neurons in layer $l-1$ which are connected to the neuron in consideration, w represents the weights of those connections, b is the bias and f is the activation function. In this way, the features in the input layer are mapped via hidden layers to the prediction in the output layer. A fully connected artificial neural network with one hidden layer is shown in Figure 2.4. There are many other varieties of neural networks as well.

2.7 Network Architecture for Emergence Prediction

Network architecture plays an important role in how information is processed and system behavior emerges (Csermely et al., 2013). The top regulators within the GRN architecture identified by K-core decomposition method can be used to effectively predict the emergent state of the regulatory system (Narang et al., 2015). The bow-tie architectural feature has been associated with emergent properties like controllability (Csete and Doyle, 2004). Increase in the bow-tie CORE size is linked with decrease in controllability. This points to the bow-tie architectural feature being a useful basis for predicting a general trend in the emergence of this property.

Emergence of system properties can be predicted by molecular network dynamical models which combine architecture and dynamical information (Tavassoly, Goldfarb, and Iyengar, 2018). Several studies have shown how network architecture and dynamics together can be used to study emergent behavior at different levels (Tyson, K. C. Chen, and Novak, 2003; Alon, 2007; J.-R. Kim, Yoon, and K.-H. Cho, 2008; Long, Brady, and Benfey, 2008; Muhammad et al., 2017; Zanudo and Albert, 2015). Aspects of the architecture like edge signs and directions and feedback loops are used in dynamical modeling. The modeling

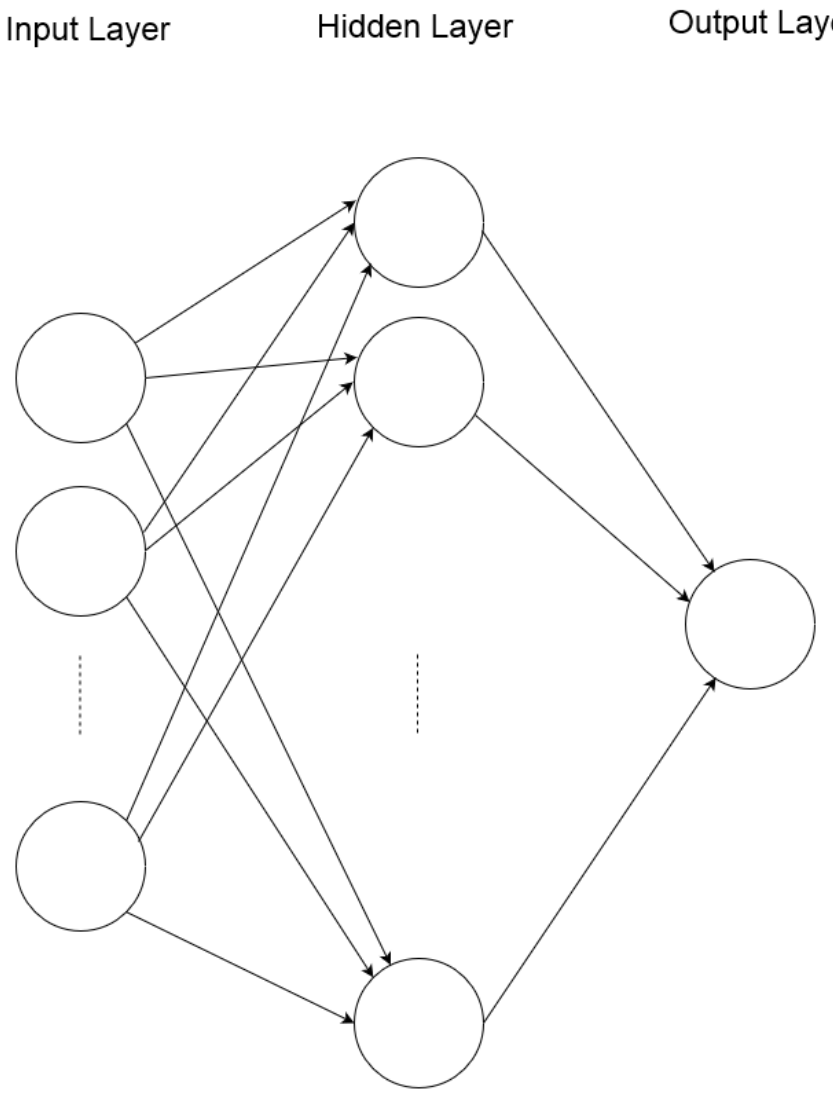


Figure 2.4: A fully connected artificial neural network with one hidden layer. The circles represent the neurons.

becomes very complicated with many parameters as the size of the network increases. When the behavior at the level of the molecular system needs to be mapped to that at higher biological levels, the number of associated parameters in the model would further increase.

The task of mapping genotype to phenotype, that is lower-level network state measurements to higher-level emergent properties, can be addressed as a learning task. An

example of organism-level phenotype is patient cancer survival risk. Machine learning models are used for the prediction of cancer survival risk using gene expression data (W.-Y. Cheng, T.-H. O. Yang, and Anastassiou, 2013), including neural networks or NNs (Katzman et al., 2018; Z. Huang et al., 2019). However one major limitation of some machine learning models including standard neural networks is lack of interpretability, as they are used as black boxes. Model interpretability refers to the degree to which the model's internal operations can be understood by a human (Biran and Cotton, 2017). A more interpretable model would inherently increase a user's trust on the model. The emergent phenotype of cancer-specific survival risk is clinically very relevant as its prediction can inform patient treatment, and user's trust is important in such a high-stakes prediction.

To increase model interpretability compared to standard black box NNs, there are visible NNs or VNNs where biological meanings are attached to intermediate neurons and connections between the neurons represent biological relationships (M. K. Yu et al., 2018). VNNs are designed using biological pathway knowledge (Elmarakeby et al., 2021; Hilten et al., 2020; J. Ma et al., 2018; Fortelny and Bock, 2020). More details of VNNs are provided in Section 5.1. However, none of the state-of-the-art VNNs model how a known pathway architecture can change for a particular disease or type of cancer. The design of VNNs is based on biological relationships, so modeling how known pathway architectures change under a type of cancer can be useful in the prediction of the emergent phenotype of cancer-specific survival risk.

2.8 Chapter Summary

In this chapter we present a literature review of the prior work and concepts which are relevant to the central objective of the thesis. We review emergent properties of biological

systems and how they can be useful in understanding of disease characteristics and development of therapeutic strategies. We review the concept of biological networks like GRNs and signaling networks, and how their architectures can be obtained from data sources. However, vital information in GRN data sources can be missing, information like that of edge signs which can be useful for the prediction of emergence. We review GRN inference algorithms and their major limitations like not predicting edge signs, or predicting signs with no edge directions or with no cycles, pointing to the need for better GRN inference from expression data. In the context of network architectural features, we review the K-core network decomposition which can denote which regulators in a GRN are most important. We review the bow-tie architectural feature, and show that prior work has investigated the existence of this feature in only some eukaryotic GRNs. We explore how the bow-tie architecture is associated with quantitative changes in emergent properties like controllability, suggesting the need of an investigation into GRNs of widely varying species complexity to predict a trend in emergence. We review the prediction of organism-level emergent phenotype from genotype as a learning task. We see that biological pathway knowledge is used in prediction with VNNs which provide increased interpretability compared to standard NNs, and that none of the VNNs model how known pathway architecture can change for particular diseases.

Chapter Three

Inference of Signed Gene Regulatory Network Architecture

In this chapter we focus on the aspect of GRN architecture that is useful in the prediction of emergence. Inferring gene regulatory networks (GRNs) from gene expression data is a significant systems biology problem. A useful inference algorithm should not only unveil the global structure of the regulatory mechanisms but also the details of regulatory interactions like edge signs denoting activation or inhibition. Many popular GRN inference algorithms cannot infer edge signs, and those that can infer signed GRNs cannot simultaneously infer edge directions or network cycles.

To address these limitations of existing algorithms we propose a novel algorithm Polynomial Lasso Bagging (PoLoBag) for signed GRN inference with both edge directions and network cycles. PoLoBag is an ensemble regression algorithm in a bagging framework where Lasso weights estimated on bootstrap samples are averaged. The bootstrap samples incorporate polynomial features to capture higher order interactions. We evaluate the signed inference performance of our algorithm against state-of-the-art algorithms on simulated and real-world expression datasets. Next we aim to validate the role of signed GRN architecture

in the prediction of emergence. We use a combination of signed GRN architecture and dynamical information in our proposed dynamical K-core method for emergent state prediction of the gene regulatory system.

The organization of this chapter is as follows. Section 3.1 introduces the concept of signed gene regulatory network inference and presents related work. We describe our proposed PoLoBag algorithm and further experimental details in Section 3.2. The signed inference results are presented in Section 3.3 with future directions discussed in Section 3.4. In Section 3.5 we present our work on the dynamical K-core decomposition method for finding top regulators and predicting emergent gene regulatory system states. A summary of this chapter is given in Section 3.6.

3.1 Background and related work

Reverse-engineering the gene regulatory network from high-throughput expression data, or network inference, is a challenging and important research area (Gardner and Faith, 2005; W.-P. Lee and Tzou, 2009). Gene regulatory networks (GRNs) are networks that represent regulatory interactions between regulators such as transcription factors, kinases, etc. and target genes. Study of these networks is key to understanding several system-level responses in the organism crucial to its development and pathology. This in turn helps in developing effective therapeutic approaches for critical diseases like cancer. Many widely used GRN inference algorithms cannot infer the nature of regulation, where activating and inhibitory interactions are represented by positive and negative signs respectively. State-of-the-art inference algorithms capable of predicting edge signs suffer from limitations like inferred networks either not having edge directions or not allowing cycles.

Edge signs are very important in regulatory network analysis. The effects in the

overall system caused by perturbations to regulators can be determined using these signs. Feedforward and feedback loops found in regulatory networks exhibit different dynamics based on the signs of the involved edges (Alon, 2007). Comprehensive understanding of GRNs includes knowing the activating/inhibitory nature of the regulatory interactions and such understanding can have biological and clinical applications. One such major application is in the area of drug development using network-based approaches (Barabási, Gulbahce, and Loscalzo, 2011). Other applications of edge signs include identification of pluripotency and differentiation related genes in murine embryonic stem cells (M. J. Mason et al., 2009).

A number of algorithms have been proposed for signed inference from expression data obtained under different experimental settings. Network identification by multiple regression (NIR) (Gardner, Di Bernardo, et al., 2003) is based on ordinary differential equations (ODEs) and requires knowledge of perturbed genes. The algorithm (Bansal, Gatta, and Di Bernardo, 2006) performs signed inference from time course gene expression data. Network Inference with Multi Objective Optimization (NIMOO) (Gupta et al., 2011) utilizes multi-objective optimization to integrate multiple inference methods and experimental data sources. Transitive Reduction and Closure Ensemble (TRaCe+) (Ud-Dean et al., 2016) constructs an ensemble of signed directed networks from gene knock-out expression data. However, we are interested in signed inference from a general form of gene expression data, like the more common and easily available multifactorial data.

Among the algorithms that can perform signed inference from expression datasets of interest, the algorithm (Veber et al., 2008) infers the role of regulators by checking for consistency of regulatory networks with signs of variation obtained from expression data. However, it needs wild-type or reference gene measurements to obtain these signs of variation. Petri Nets with Fuzzy Logic (PNFL) (Küffner et al., 2010) performs signed inference from diverse datasets by modeling GRNs as PNFL. This algorithm also requires wild-type gene measurements before PNFL simulation. Two readily available state-of-the-art algorithms

without the requirement of reference measurements are Bayesian Network Inference with Java Objects (Banjo) (J. Yu et al., 2004) and Signing of Regulatory Networks (SIREN) (Khosravi et al., 2015).

Banjo belongs to the category of probabilistic graphical model based algorithms. In Banjo the probabilistic relationships between variables are represented using Bayesian networks. The best network in terms of fitting the observed data is found by a search. An influence score is computed for network edges denoting the magnitudes and signs of interactions. Banjo can capture many forms of relationships between variables and handle noisy data because of its probabilistic nature. However, the algorithm requires a lot of data for accurate inference. In addition static Bayesian networks in Banjo cannot represent cycles like feedback loops which are often observed in regulatory networks. Version 2 of Banjo software can create a consensus graph from the best fit networks, which can have cycles, but signed edge scores are not available for this consensus graph.

SIREN algorithm is from the category of information theoretic algorithms. Signed weighted gene co-expression network analysis (WCGNA) (M. J. Mason et al., 2009) uses an information theoretic similarity metric based on Pearson correlation coefficient (PCC) to generate signed networks. Nonlinear relationships between variables cannot be captured by correlation measures, so mutual information is used for network inference. However, mutual information is a non-negative quantity. SIREN overcomes this limitation by using a mutual information-based measure with a rescaling matrix to produce signed scores. The rescaling matrix is used to differentiate the expression distribution pattern of a pair of positively correlated genes from the pattern of a pair of negatively correlated genes. Working in a complementary way with other inference algorithms that can produce an unsigned GRN, SIREN can deduce the edge signs. It is fast and quite accurate with reasonable amounts of data and is shown to be better than PCC on real-world datasets. But the major disadvantage of this algorithm used on its own is that inferred signed interactions are not directional.

There are some algorithms for signed inference from expression data of our interest which use Least absolute shrinkage and selection operator (Lasso) (Tibshirani, 1996). Inferelator (Bonneau et al., 2006) uses an ODE model and applies Lasso for variable selection, and also considers nonlinear interactions involving more than one regulator. As a pre-processing step, genes and conditions are first grouped into biclusters using a biclustering algorithm. Lasso regression has been used for the task of finding signed edge weights for some target genes in a gene network (Licausi et al., 2011). However, Lasso on its own suffers from limitations where the variable selection is firstly constrained by the sample size and secondly is not accurate when there are highly correlated input variables (Zou and T. Hastie, 2005). The algorithm (Gustafsson et al., 2009) uses an ODE model with Lasso for variable selection. A bootstrap procedure of data sampling is used for obtaining edge scores. The bootstrapping framework in (Morgan et al., 2019) can be used with Lasso to perform signed inference. However, feature bagging which can address the second aforementioned limitation of Lasso (Sijian Wang et al., 2011) is not used in these two algorithms.

3.2 Methods

We propose Polynomial Lasso Bagging (PoLoBag), an ensemble regression algorithm for signed GRN inference from gene expression data. By utilising a regression setting with input and output variables, network edges have direction (from input to output) unlike SIREN. Separate regression problems for each target gene in this setting allows for cycles or loops unlike Banjo. A bagging ensemble framework (Breiman, 1996) is used in this algorithm. Each model in the ensemble is a Lasso model which works as a simple variable selector estimating the connection weights between input and output variables. Most importantly in the Lasso model these weights are signed. Signed weights from each Lasso model trained on a separate bootstrap sample are aggregated to produce the signed edge weights. Unlike

Inferelator, PoLoBag does not use a separate ranking method to initially select a smaller number of highest confidence regulators for every target gene, and the bagging framework in PoLoBag incorporates both data sampling and feature bagging. This bagging framework was inspired by the Random Lasso method (Sijian Wang et al., 2011) to address critical limitations of Lasso. However, unlike Random Lasso, in PoLoBag each bootstrap sample uses polynomial features. These can capture higher order interactions which are expected to be observed more frequently in GRNs of more complex organisms.

The PoLoBag algorithm is presented in Figure 3.1. Let $\boldsymbol{D} \in \mathbb{R}^{n \times m}$ denote the input gene expression data for n genes and m measurement conditions. With n_R potential network regulators, the objective of signed inference is to find the vector $\boldsymbol{w} \in \mathbb{R}^{n_R(n-1)\times 1}$ comprised of signed edge weights between the regulators and target genes in the underlying network with no autoregulation. These weights represent the strength and nature (activating/inhibitory) of regulatory interactions. With no prior knowledge of regulators, we will consider all genes to be potential network regulators with $n_R = n$. PoLoBag is an ensemble regression algorithm where the network inference problem is divided into a separate regression task for each target gene. Each regression task is performed using an ensemble of Lasso models in a bagging framework. Each Lasso model is trained on a bootstrap sample created by selecting measurement conditions randomly with replacement. The term sample (measurement sample) could also refer to each individual measurement condition, here we use the term sample to refer to a set of such measurement conditions. Each bootstrap sample incorporates a random set of polynomial features. The Lasso coefficients estimated from each bootstrap sample are averaged to produce the corresponding signed weights in $\boldsymbol{w}$.

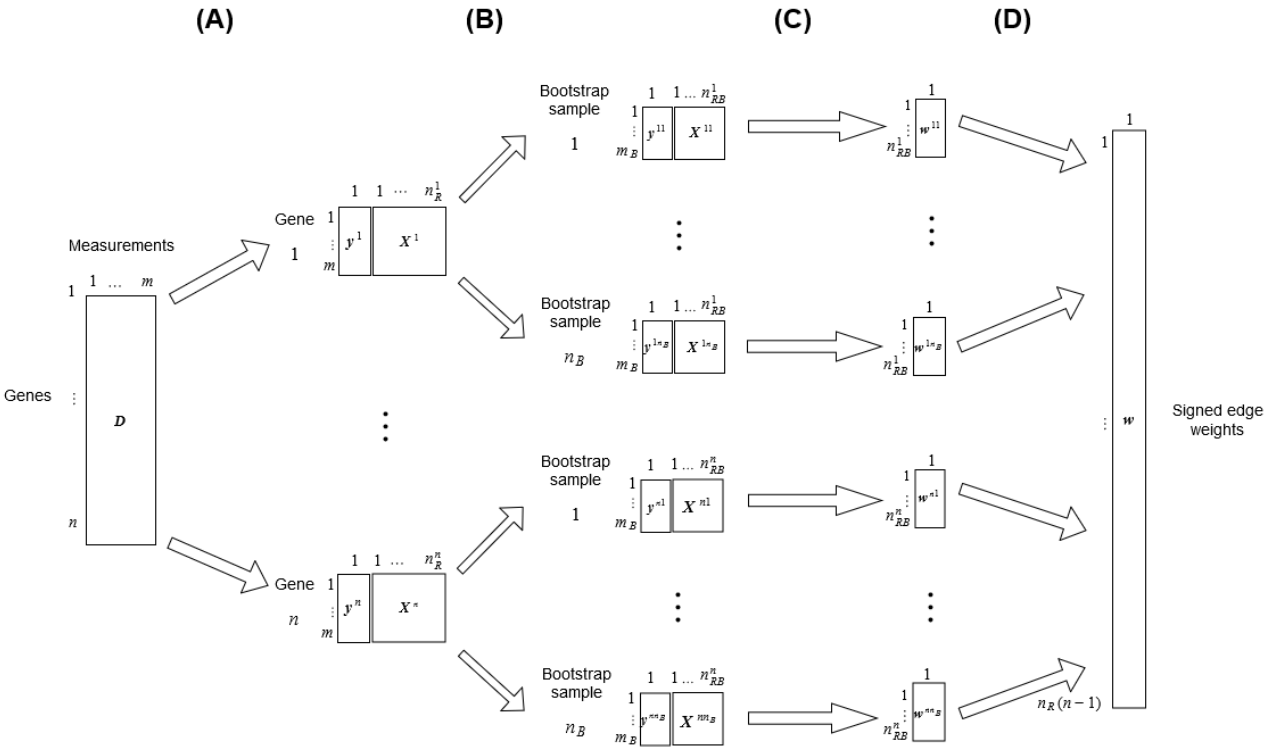


Figure 3.1: Overview of PoLoBag. (A) The input gene expression data $\boldsymbol{D} \in \mathbb{R}^{n \times m}$ where n is the number of genes and m is the number of measurement conditions is separated into input and output variables for each target gene t, $t = 1, 2, \cdots, n$. (B) For each input-output pair $(\boldsymbol{y}^t, \boldsymbol{X}^t)$, bootstrap samples are generated. Each sample consists of n^t_{RB} random polynomial features with m_B random measurements conditions. (C) Lasso weights are estimated from each bootstrap sample. (D) These weights are averaged over all samples in the bagging framework and this process is repeated for all target genes to produce the signed weights for $n_R(n-1)$ possible network edges, where n_R is the number of potential network regulators.

3.2.1 Pre-processing

In this algorithm we want the Lasso regression coefficients to have positive signs for activating interactions and negative signs for inhibitory interactions between input and output variables. This can be enabled by representing high expression values as positive and low expression values as negative numbers. Additionally we want the edge weights estimated for each target gene separately to be comparable across all the genes even if the expression values for different genes are in different ranges. So as a pre-processing step, we convert the expression profile for every gene to Z-scores computed across all measurement conditions. Initial exploration of the expression data after setting the mean to zero with Z-score normalization additionally revealed that the distribution of values was skewed. Expression values equally separated into positive and negative numbers in a neighborhood of zero can help in better deducing the sign of the underlying interactions. This sign separation should capture the maximum number of expression values, close in value to each other. So we shift the data by τ times its median (line 6 in Algorithm 1) before regression.

3.2.2 Lasso

Lasso acts as a variable selector by selecting input variables or features that can best explain the output variable. The corresponding input-output weights that are estimated can have signs. For a sample dataset b let vector $\boldsymbol{y}^{tb} \in \mathbb{R}^{m_B \times 1}$ represent the expression of target gene t in m_B measurement conditions. Let the matrix $\boldsymbol{X}^{tb} \in \mathbb{R}^{m_B \times {n^t}_{RB}}$ denote the corresponding ${n^t}_{RB}$ input features. The relationship between input-output variables in this model is given by

$$\boldsymbol{y}^{tb} = \boldsymbol{X}^{tb}\boldsymbol{w}^{tb} + \boldsymbol{\epsilon}^{b}, \tag{3.1}$$

Algorithm 1 PoLoBag algorithm.

Input: Gene expression data $\boldsymbol{D}$, **Output**: Signed edge weights $\boldsymbol{w}$

Convert $\boldsymbol{D}$ to Z-scores;

$\boldsymbol{w} \leftarrow$ Initialize to empty;

foreach target gene $t \in \{1, \cdots, n\}$ **do**

  $\boldsymbol{y}^t, \boldsymbol{X}^t \leftarrow$ Expression profiles of target gene, potential regulators;

  Subtract $\tau median([\boldsymbol{y}^t, \boldsymbol{X}^t])$ from $\boldsymbol{y}^t$ and $\boldsymbol{X}^t$;

  $\boldsymbol{w}_M^t, \boldsymbol{w}_S^t, \boldsymbol{s}_w^t \leftarrow$ Initialize to zero vectors;

  $\boldsymbol{n}_F^t \leftarrow$ Put $n_d^k \sqrt{n_R^t} \forall k \in \{1, \cdots, d\}$

  foreach $b \in \{1, \cdots, n_B\}$ **do**

    $\boldsymbol{y}^{tb}, \boldsymbol{X}^{tb}, \boldsymbol{id}^{tb} \leftarrow$ BOOTSTRAPSAMPLE ($\boldsymbol{y}^t$,$\boldsymbol{X}^t$,n_M,$\boldsymbol{n}_F^t$);

    Fit a Lasso model on $\boldsymbol{y}^{tb}, \boldsymbol{X}^{tb}$ and obtain $\boldsymbol{w}^{tb}$;

    Update $\boldsymbol{w}_M^t, \boldsymbol{w}_S^t, \boldsymbol{s}_w^t$ using Eq. 3.8;

  end foreach

  Compute $\boldsymbol{w}^t$ using Eq. 3.4 and put in $\boldsymbol{w}$;

end foreach

function BOOTSTRAPSAMPLE($\boldsymbol{y}^t$,$\boldsymbol{X}^t$,n_M,$\boldsymbol{n}_F^t$)

  $\boldsymbol{y}^{tb}$,$\boldsymbol{X}_F^{tb} \leftarrow$ Select $n_M m$ random rows from $\boldsymbol{y}^t$,$\boldsymbol{X}^t$;

  $\boldsymbol{X}^{tb}, \boldsymbol{id}^{tb} \leftarrow$ Initialize to empty;

  foreach $k \in \{1, \cdots,$ Length of $\boldsymbol{n}_F^t\}$ **do**

    Select $\boldsymbol{n}_F^{tk}$ random unique new $\boldsymbol{X}_F^{tb}$ cols in each of $\boldsymbol{X}_F^{tbk1}, \cdots, \boldsymbol{X}_F^{tbkk}$;

    foreach $c \in \{1, \cdots, 2^{(k-1)}\}$ **do**

      Compute $\boldsymbol{F}^{tbkc}$ using Eq. 3.6;

      Append c^{th} set of $\frac{\boldsymbol{n}_F^{tk}}{2^{(k-1)}}$ cols from $\boldsymbol{F}^{tbkc}$ to $\boldsymbol{X}^{tb}$;

      For each appended $\boldsymbol{F}^{tbkc}$ col put $\boldsymbol{X}_F^{tb}$ col indices,c in $\boldsymbol{id}^{tb}$;

    end foreach

  end foreach

  return $\boldsymbol{y}^{tb}, \boldsymbol{X}^{tb}, \boldsymbol{id}^{tb}$;

end function

where $\boldsymbol{w}^{tb}$ is the vector of signed connection weights between input and output variables and $\boldsymbol{\epsilon}^{b}$ represents random noise. No intercept term is used here. Lasso regression is a linear model with L1 sparsity prior as the regularizer. The vector $\boldsymbol{w}^{tb}$ is obtained by minimizing

$$\frac{1}{2m_B}\|\boldsymbol{y}^{tb} - \boldsymbol{X}^{tb}\boldsymbol{w}^{tb}\|_2^{(2)} + \alpha\|\boldsymbol{w}^{tb}\|_1, \tag{3.2}$$

where α is the Lasso regularization parameter that controls sparsity.

3.2.3 Bagging

Lasso can obtain signed coefficients but suffers from some limitations in practice (Zou and T. Hastie, 2005). First, the number of selected variables is limited by the number of data points. Second, when there are several highly correlated input variables related to the output variable, Lasso tends to select only a few of these input variables. The latter limitation can become a problem for regulatory networks where many regulator genes in synergy can control a target gene. The expression profiles of these regulators would probably have higher correlation than those of any two randomly selected genes in the network, and Lasso might select only a few of these regulator genes. A bagging framework with data sampling and feature bagging similar to the first step of the two step Random Lasso algorithm is able to circumvent these limitations.

Data sampling

Each individual Lasso in this framework trains on a bootstrap sample dataset with the measurement conditions chosen via random sampling with replacement. The size of each of the n_B bootstrap samples is given by

$$m_B = n_M m, \tag{3.3}$$

where n_M selects a fraction of the total number of measurements.

Feature bagging and polynomial features

The features used in the bootstrap sample are selected randomly without replacement. All the relevant input features for the output variable might not be there in a sample. Different samples additionally have different sets of measurement conditions. As such it is possible that Lasso selects the same relevant input feature with high weight magnitudes but with different signs from different samples. In such a case using the average of these Lasso weights would produce a reduced magnitude. So this algorithm uses a scheme where the weight magnitude and sign are separated in column vectors $\boldsymbol{w}_M^t \in \mathbb{R}^{n_R^t \times 1}$ and $\boldsymbol{w}_S^t \in \mathbb{R}^{n_R^t \times 1}$ for target gene t with n_R^t potential regulators. With no autoregulation, for a target gene, $n_R^t = n_R - 1$ if it itself is a potential network regulator or $n_R^t = n_R$ otherwise. The vector $\boldsymbol{s}_w^t \in \mathbb{R}^{n_R^t \times 1}$ stores the total number of times each regulator expression profile is used in features across all samples. These vectors are updated by each Lasso model and combined at the end to produce edge weights $\boldsymbol{w}^t$ for target gene t by

$$\boldsymbol{w}^t = \frac{sign(\boldsymbol{w}_S^t)\boldsymbol{w}_M^t}{\boldsymbol{s}_w^t}. \tag{3.4}$$

This is done for every target gene t to obtain $\boldsymbol{w}$.

The PoLoBag algorithm incorporates polynomial features which represent not only first order interactions (linear features) but also higher order multiplicative interactions (non-linear features). It is important to consider these multiplicative interactions in regulatory networks where regulators in synergy control the expression of the target gene. We can create such nonlinear features in a sample by multiplying individual input features, however this poses some issues. For Lasso weights comparable across all used features in a bootstrap sample, first we want the nonlinear features to have values in the same range as linear ones. Second, from the estimated sign of a nonlinear feature weight, the signs corresponding to the

individual input features involved need to be deduced. Keeping these issues in consideration, the polynomial features comprised of both linear and nonlinear features in a bootstrap sample are constructed in the BOOTSTRAPSAMPLE function in Algorithm 1.

In PoLoBag algorithm with a defined polynomial degree d, we create polynomial features by combining k unique individual input features, where $k \in \{1, \cdots, d\}$. Let n_d^k represent d user defined values to control the number of polynomial features for every k. To create these features for a given k, we select $n_d^k\sqrt{n_R^t}$ random unique individual input feature columns to be put in each of $\boldsymbol{X}_F^{tbk1}, \cdots, \boldsymbol{X}_F^{tbkk} \in \mathbb{R}^{n_M m \times n_d^k\sqrt{n_R^t}}$. Here we do not reuse an individual feature that has been already used in the bootstrap sample, however it can be used in other bootstrap samples. For easy deduction of individual signs from Lasso weights, the idea is to have different feature categories each of which would only represent polynomial interactions of a particular form. This form is defined in terms of the sign of one fixed individual feature, for instance the one in $\boldsymbol{X}_F^{tbk1}$ and whether the signs of all other $k-1$ individual features each taken from the same column position of $\boldsymbol{X}_F^{tbk2}, \cdots, \boldsymbol{X}_F^{tbkk}$ are same or different in comparison in the represented interaction. Accordingly we define a matrix $\boldsymbol{S} \in \{-1, +1\}^{2^{(k-1)} \times k}$ where the first column consists of all 1s and the remaining $k-1$ columns are all possible distinct ± 1 variations.

$$\boldsymbol{S} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & -1 & 1 & \cdots & 1 \\ 1 & 1 & -1 & \cdots & 1 \\ & & \vdots & & \\ 1 & -1 & -1 & \cdots & -1 \end{bmatrix} \tag{3.5}$$

Polynomial features for a given k in categories $c \in \{1, \cdots, 2^{(k-1)}\}$ are created as

$$\begin{aligned}\boldsymbol{A}^{tbk} &= \frac{1}{2^{(k-1)}}|\boldsymbol{X}_F^{tbk1}\cdots\boldsymbol{X}_F^{tbkk}|^{(\frac{1}{k})},\\ \boldsymbol{F}^{tbkc} &= sgn(\boldsymbol{X}_F^{tbk1})\left[\prod_{i=2}^{k}\left[1+\boldsymbol{S}_{ci}sgn(\boldsymbol{X}_F^{tbk1}\boldsymbol{X}_F^{tbki})\right]\right]\boldsymbol{A}^{tbk},\end{aligned} \tag{3.6}$$

where sgn represents the mathematical sign function. In each bootstrap sample, polynomial features for a given k are selected in equal number from each of the c categories. For a given category, we select $\frac{1}{2^{(k-1)}}$ fraction of all $n_d^k\sqrt{n_R^t}$ feature columns from $\boldsymbol{F}^{tbkc}$, where the selected column indices are unique for every category c (line 23 in Algorithm 1). Overall the total number of features in a bootstrap sample is

$$n_{RB}^t = \sum_{k=1}^{d} n_d^k\sqrt{n_R^t}. \tag{3.7}$$

To update weight magnitude and sign vectors, the individual input feature indices and the category for every feature selected in a bootstrap sample is stored in $\boldsymbol{id}^{tb}$. For an estimated Lasso weight $\boldsymbol{w}^{tbl}$ in $\boldsymbol{w}^{tb}$ where $l \in \{1,\cdots,n_{RB}^t\}$, let the corresponding $\boldsymbol{id}^{tbl}$ give k individual feature indices combined to create the polynomial feature and the category c. The weight magnitude and sign vectors are updated $\forall i \in \{1,\cdots,k\}$

$$\begin{aligned} f &= \boldsymbol{id}^{tbli},\\ \boldsymbol{w}_M^{tf} &= \boldsymbol{w}_M^{tf} + |\boldsymbol{w}^{tbl}|^{(\frac{1}{k})},\\ \boldsymbol{w}_S^{tf} &= \boldsymbol{w}_S^{tf} + \boldsymbol{S}_{ci}sgn(\boldsymbol{w}^{tbl})|\boldsymbol{w}^{tbl}|^{(\frac{1}{k})},\\ \boldsymbol{s}_w^{tf} &= \boldsymbol{s}_w^{tf} + 1.\end{aligned} \tag{3.8}$$

This polynomial feature creation framework in PoLoBag is generalized here for all $k \in \{1,\cdots,d\}$ for any d. So both linear and nonlinear polynomial features can be created. Though higher values of d could capture higher order network interactions, we use a value of $d = 2$ considering the associated complexity.

3.2.4 Parameters

In the PoLoBag algorithm with degree $d = 2$ there are five parameters.

- n_2^1 and n_2^2 - These are the two user defined parameters in Equation 3.7 for $k = 1$ and $k = 2$. These control the number of linear and nonlinear polynomial features selected in each bootstrap sample respectively. For ease of user selection these values are fractions that get scaled by the square root of the number of potential regulators of the target gene.
- n_M - This controls the bootstrap sample size. In equation 3.3 this parameter value as a fraction gets multiplied by the total number of measurement conditions to give the size of each bootstrap sample.
- n_B - This denotes the total number of bootstrap samples in the ensemble.
- α - The regularization parameter balances sparsity with data fit in each Lasso model.

The value of τ can be tuned to control data shift, here we set a value of $\tau = 3$. The default values of other standard Lasso parameters are used. Depending on the nature of the underlying regulatory mechanisms and data, these five parameters influence PoLoBag's inference performance differently (Section 3.3). However, with the same set of experimentally chosen values, PoLoBag achieves accurate results across nine diverse test datasets.

3.2.5 Experiments

Datasets

The experimental datasets are listed in Table 3.1. The *in silico* data used is multifactorial data, which are steady-state measurements obtained from applying multifactorial perturba-

Dataset	Type	Organism	Genes	Directed	Positive edges	Negative edges	Unknown sign edges	Measurements/gene
A	Simulated	Yeast	200	Yes	238	237	0	200
B	Simulated	Yeast	400	Yes	532	539	0	400
C	Simulated	Yeast	500	Yes	872	950	0	500
D	Simulated	*E. coli*	500	Yes	812	568	0	500
E	Simulated	*E. coli*	650	Yes	886	637	0	650
F	Real	*E. coli*	1419	No	1408	1279	0	907
G	Real	Human	522	No	175	102	1155	171
H	Real	*E. coli*	99	Yes	144	80	0	24
I	Real	Human	408	Yes	1283	762	0	200

Table 3.1: Experimental datasets for signed inference.

tions to the underlying network. These perturbations are simulated by slightly increasing or decreasing the basal activation of all network nodes by random amounts. These were generated using the tool GeneNetWeaver or GNW (Schaffter, Marbach, and Floreano, 2011; Marbach, Schaffter, Mattiussi, et al., 2009). The GNW data generation settings were those used in DREAM4 *In Silico Multifactorial* subchallenge – coefficient of the molecular noise in the stochastic simulation = 0.05, the measurement noise as a mix of normal (standard deviation = 0.025) and lognormal (standard deviation = 0.075) noise, and normalization after adding the measurement noise.

We used two real datasets previously used by the authors of SIREN (Khosravi et al., 2015), where the interaction directions are not considered. One is for the subnetwork extracted from the *E. coli* GRN in RegulonDB database (Gama-Castro et al., 2008). The gene expression data was from the Many Microbe Microarray Database M^{3D} (Faith et al., 2007). The other dataset is for a prostate cancer GRN extracted by SIREN authors from the STRING functional interaction database (Snel et al., 2000). The data was collected from the GEO database with accession number GDS2545. The measurement samples correspond to gene expression measurements in four cell states - normal prostate tissue, normal prostate tissue adjacent to tumor, primary prostate tumor tissue and metastatic prostate cancer tissue (Chandran et al., 2007). In our experiments, we used the 526 genes of SIREN authors for

which this expression data was found available, resulting in a slightly smaller subnetwork consisting of 522 genes.

We have another two real test datasets for directed networks. One is for a subnetwork extracted from *E. coli* GRN obtained from RegulonDB database (Santos-Zavaleta et al., 2019). The expression data was obtained from the GEO database with accession number GSE135516. *E. coli* evolution was performed to study adaptation to iron toxicity (Anand et al., 2020). The second dataset is for a subnetwork extracted from human GRN in TRRUST v2 database (H. Han et al., 2018). The expression data, obtained from the GEO database with accession number GDS3795, is from bone marrow CD34+ cells of myelodysplastic syndrome patients and healthy controls (Pellagatti et al., 2010). In both cases, considering primarily ground truth edges having positive or negative signs defined with high confidence, we extracted subnetworks of interest consisting of high degree nodes connected with each other as a weakly connected component. The datasets and the PoLoBag code are available at https://github.com/gourabghoshroy/PoLoBag.

3.2.6 Performance evaluation

The signed inference performance was assessed by building on the metrics used in SIREN paper. Let the total number of true network edges be E. Some of these might not have known signs in the ground truth. Retrieval denotes the fraction of true edges signed by the inference algorithm. The metric recall is defined as the fraction of relevant instances that are retrieved. So here recall is the same as the retrieval when the relevant instances are signed true edges. Let SE represent the number of true edges that are signed by the algorithm, then the fraction recall is defined as

$$\text{Recall} = \frac{SE}{E}. \tag{3.9}$$

In signed inference we want to know how accurate the inferred signs are. Hence in addition we use the metric of accuracy which denotes the fraction of the known sign true edges that are signed correctly by the algorithm. Let PTP and NTN represent the number of positive and negative true edges whose signs are correctly inferred to be positive and negative respectively. Here P and N mean the positive and negative signs of the edges. Let NFP be the true edges whose signs are negative but are inferred incorrectly to be positive by the algorithm. PFN similarly denotes the number of positive edges where the algorithm incorrectly predicts the signs to be negative. Accuracy is defined as

$$\text{Accuracy} = \frac{PTP + NTN}{PTP + NTN + NFP + PFN}. \tag{3.10}$$

The accuracy is plotted against the recall values to generate the accuracy-recall curve. We consider the point $(0, 0)$ to be part of the curve. In this work we used the area under the accuracy-recall curve (AUAR). It is a value in $[0, 1)$ where a higher value denotes better signed inference performance. For unsigned inference where positive/negative refers to the presence/absence of an edge, the area under the precision-recall curve would be appropriate. For this signed inference setting, the accuracy of the signs of the inferred edges also needs to be considered, making this AUAR metric a better choice.

The output edge lists obtained from inference algorithms are first sorted in the decreasing order of their absolute weights. Then this list is traversed one edge at a time to compute the accuracy and recall values. There can be cases where the accuracy-recall curve is incomplete as maximum recall obtained from traversing the full output list is less than 1. After plotting the values for the edges with non-zero weights in the sorted output list, the evaluation process assigns a random sign independently to every remaining edge in the ground truth that is not in the output list or has a zero weight in the list, to complete the rest of the curve. The area under the complete accuracy-recall curve is computed. This random assignment is repeated 50 times and the average AUAR value is considered to

be the representative metric of signed inference average accuracy. This area-based performance assessment with random assignment for missing edges is motivated by the assessment methodology for unsigned network inference in DREAM challenges (Prill et al., 2010).

Our evaluation methodology considers the directions of edges while comparing against ground truth for directed networks. For SIREN algorithm, we consider all possible directed edges and infer their signed weights. For undirected networks the evaluation methodology computes recall and accuracy values as given in Equation 3.9, 3.10 but from undirected interactions. Incomplete curves are also completed similarly. The inference algorithm output list here will have edges with defined directions, so there can be edges in both directions for a pair of genes. For an undirected interaction between the pair of genes in the ground truth, the sign of the directed edge which ranks higher in the sorted algorithm output edge list is considered while plotting the accuracy-recall curve.

3.3 Results

3.3.1 Performance Assessment

In this section we present the summarized performance assessment of our PoLoBag algorithm. Inference performance of PoLoBag depends on the algorithm's parameter values. For parameter optimization, we used the datasets G, A, H and I. The performance variation with parameter settings is presented in the Section 3.3.2. In each experiment one of the 5 important parameter values discussed in Section 3.2.4 was varied. The parameters n_2^1 and n_2^2 were varied together to also show the impact of the ratio of linear and nonlinear features. In Section 3.3.3 we also demonstrate that PoLoBag performs better overall, specially for real datasets, when there is pre-processing data shift and no intercept in the model. We use these

experiments to find optimal parameter values.

Our parameter settings experiments reveal that when the number of nonlinear features is larger than that of linear features, better performance is obtained for regulatory networks of more complex organisms like humans. A higher percentage of multiplicative interactions might exist in these networks and $n_2^2 > n_2^1$ is a better choice. With $n_2^2 < n_2^1$ better results are produced for networks of less complex organisms, where these multiplicative interactions might not be as frequent, which can also be affected by network size. The parameter settings experiments suggest that setting the bootstrap sample size to half of the total number of measurement conditions with n_M is a reasonable choice. Though a higher n_M value gives better performance especially when the number of measurements per gene is lower than the number of genes, a lower value leads to lower computational cost. With a low n_B a larger variation in the results is observed. A sufficiently large number makes the performance more stable. A low α value always gives poor results. A value in the test range was chosen to be optimal, though in some cases a higher α gives better results.

For fairness in performance comparison, we finally used in all the 9 test datasets the same set of PoLoBag parameter values selected from these parameter settings experiments - $n_2^1 = 0.5, n_2^2 = 3.5, n_M = 0.5, n_B = 500, \alpha = 0.1$. These values led to reasonably high accuracy results across all datasets. It is possible to obtain better results by varying the algorithm parameters for each of the datasets separately. For Banjo algorithm many of the same default parameter values as provided in the example settings file of the available Java code were used across all experimental datasets. Four parameters controlling run time and memory usage were modified based on the dataset size and type and available resources. The default parameters in the SIREN Cytoscape plugin (Montojo et al., 2015) were used for all the datasets. These parameter settings are listed in Section 3.3.4.

Table 3.2 shows the AUAR metric values on the benchmark datasets for PoLoBag

compared against Banjo and SIREN. The non-deterministic PoLoBag and Banjo algorithms were run 30 times and the mean and standard deviation from those independent runs are presented. PoLoBag average metric values are higher by about $0.3 - 0.4$ than Banjo and about $0.01 - 0.06$ than SIREN across all the simulated datasets. For the real datasets, the metric values are lower. This is partly due to the number of measurements per gene being lower than the number of genes, as shown in Section 3.3.5 by reducing the percentage of measurement conditions in simulated dataset C. This is also because of factors involving incompleteness and noise in the real-world expression data and the ground truth and that real regulatory mechanisms are more complex than simulation models. Still the difference in metric values between PoLoBag and the other two algorithms is significant, as detailed in Section 3.3.6. Banjo can perform better when the number of measurements per gene is much larger than the number of genes. SIREN has much higher AUAR values compared to Banjo, however PoLoBag produces the best results for all the simulated and real-world datasets across GRNs of different sizes and organisms.

In Section 3.3.7 we present the unsigned directed performance comparison between PoLoBag and GENIE3 (Huynh-Thu, Irrthum, et al., 2010), which uses a similar bagging approach with regression trees and can deal with nonlinear interactions but does not predict edge signs. Though GENIE3 performs much better on simulated datasets, PoLoBag outperforms GENIE3 on the two real datasets H and I. To illustrate how PoLoBag overcomes limitations of existing signed inference algorithms, we present the average inference on a selected subnetwork of five genes from the dataset E network (Figure 3.2). Selecting an optimal threshold for inferring the network from algorithm output is an important problem on its own. We used a set of thresholds (described in Section 3.3.8) and each algorithm's best inferred subnetwork is displayed here. PoLoBag correctly infers the positive edge between genes *flhC* and *fliM* with its direction, and the cycle between genes *ihfA* and *ihfB*.

Dataset	Banjo	SIREN	PoLoBag
A	0.5284±0.0077	0.8927	**0.9498±0.0015**
B	0.5525±0.0062	0.9035	**0.9283±0.0028**
C	0.5676±0.0057	0.8996	**0.9200±0.0026**
D	0.5561±0.0046	0.8433	**0.8612±0.0021**
E	0.5443±0.0055	0.9105	**0.9201±0.0024**
F	0.5008±0.0037	0.5365	**0.5418±0.0020**
G	0.5002±0.0122	0.6593	**0.6755±0.0088**
H	0.5010±0.0134	0.5842	**0.5929±0.0073**
I	0.5006±0.0030	0.5925	**0.6251±0.0023**

Table 3.2: Signed inference AUAR (Area under the accuracy-recall curve) metric values. For PoLoBag and Banjo the mean and standard deviation from 30 independent runs are given. The PoLoBag parameters were selected from parameter settings experiments on datasets G, A, H and I. Default parameter values were used for SIREN and Banjo. Four Banjo parameters controlling run time and memory usage were modified based on the dataset size and type and available resources, presented in Section 3.3.4.

3.3.2 Parameter Settings Experiments for PoLoBag Algorithm

We present the variation in performance of the PoLoBag algorithm with change in parameter values on four of the test datasets. The AUAR values from 30 independent runs for each setting are presented in box plots.

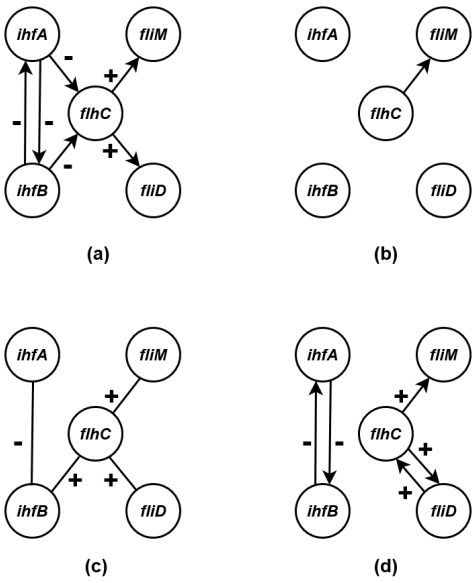
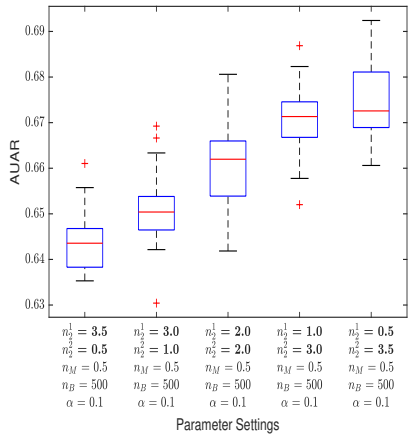
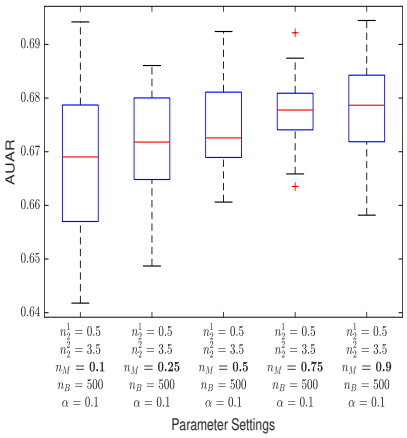


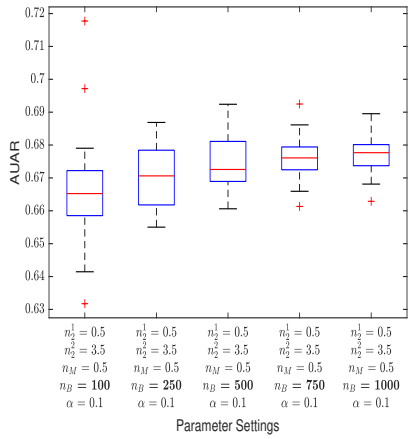
Figure 3.2: PoLoBag compared to Banjo and SIREN on part of the *E. coli* network for dataset E. (a) Ground truth. Optimal subnetworks inferred by (b) Banjo (c) SIREN (d) PoLoBag. Threshold selection is presented in the Section 3.3.8.



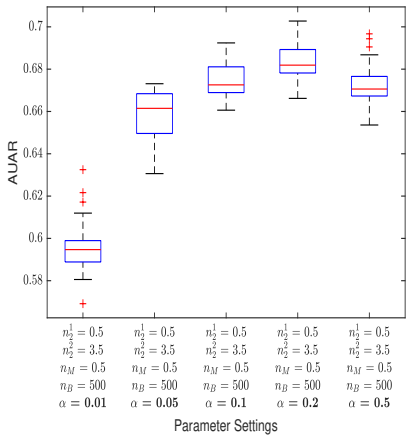
(a) Effect of parameters n_2^1 and n_2^2 on PoLoBag performance for dataset G.



(b) Effect of parameter n_M on PoLoBag performance for dataset G.

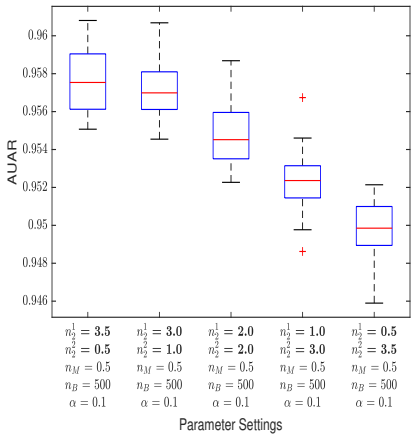


(c) Effect of parameter n_B on PoLoBag performance for dataset G.

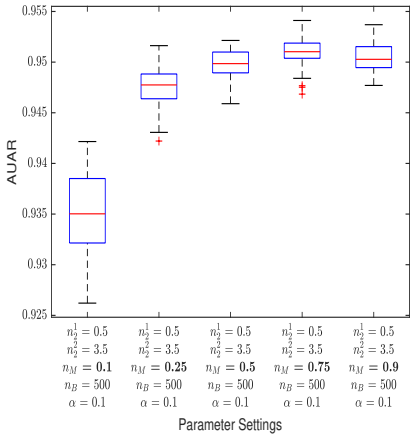


(d) Effect of parameter α on PoLoBag performance for dataset G.

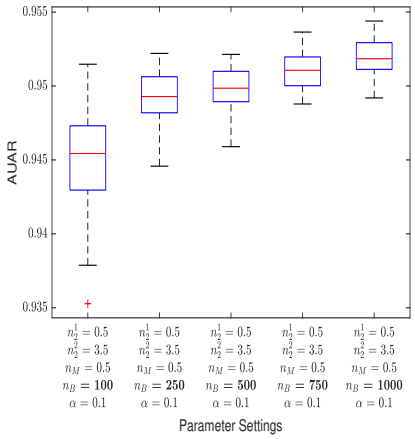
Figure 3.3: Effect of parameters on PoLoBag performance for dataset G.



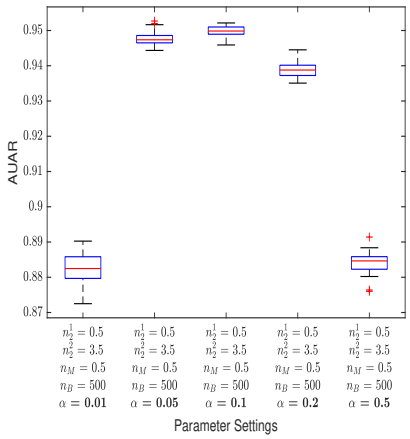
(a) Effect of parameters n_2^1 and n_2^2 on PoLoBag performance for dataset A.



(b) Effect of parameter n_M on PoLoBag performance for dataset A.

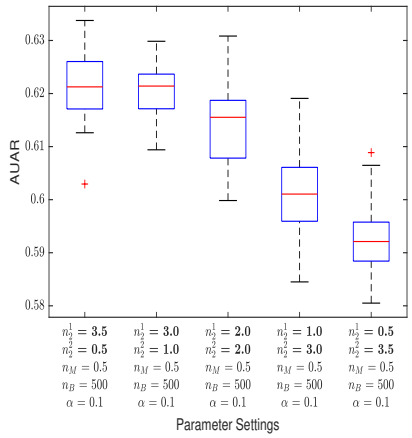


(c) Effect of parameter n_B on PoLoBag performance for dataset A.

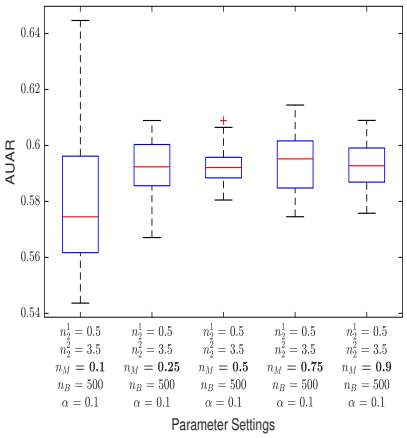


(d) Effect of parameter α on PoLoBag performance for dataset A.

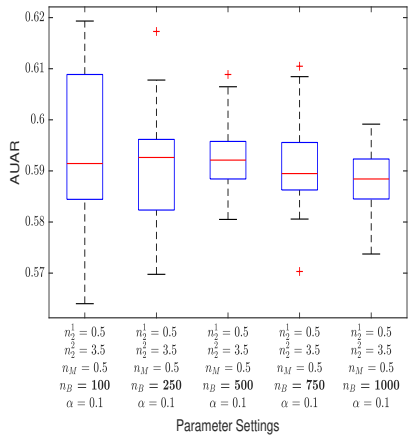
Figure 3.4: Effect of parameters on PoLoBag performance for dataset A.



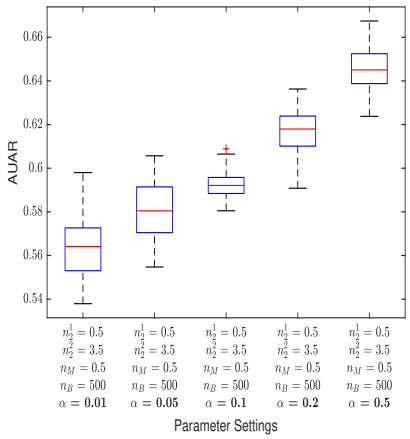
(a) Effect of parameters n_2^1 and n_2^2 on PoLoBag performance for dataset H.



(b) Effect of parameter n_M on PoLoBag performance for dataset H.

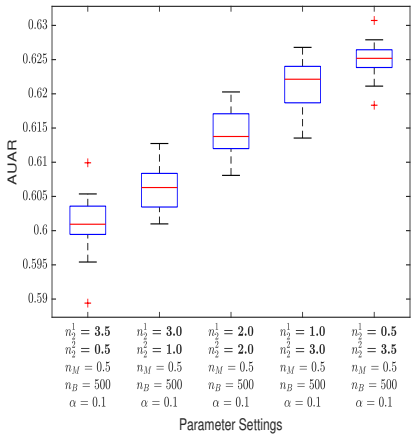


(c) Effect of parameter n_B on PoLoBag performance for dataset H.

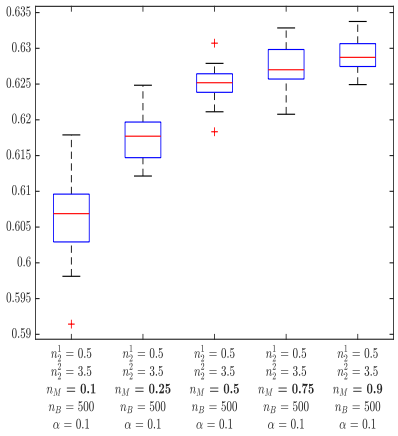


(d) Effect of parameter α on PoLoBag performance for dataset H.

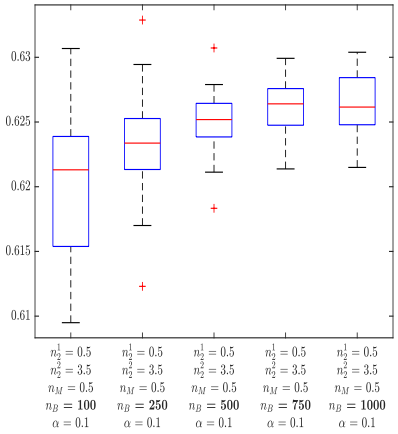
Figure 3.5: Effect of parameters on PoLoBag performance for dataset H.



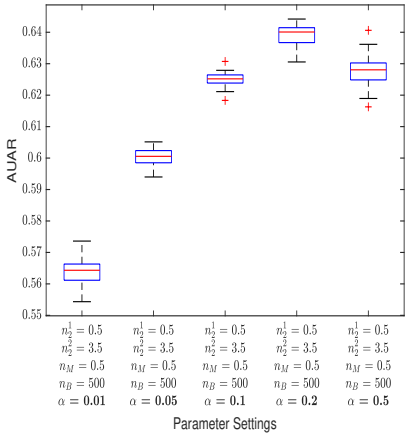
(a) Effect of parameters n_2^1 and n_2^2 on PoLoBag performance for dataset I.



(b) Effect of parameter n_M on PoLoBag performance for dataset I.



(c) Effect of parameter n_B on PoLoBag performance for dataset I.

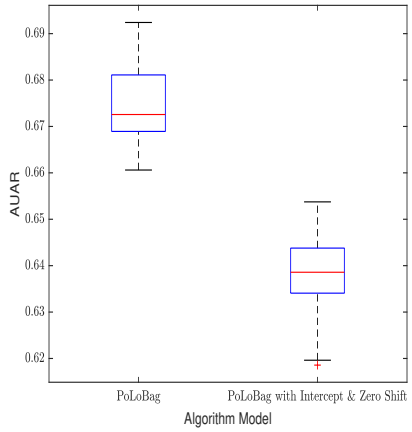


(d) Effect of parameter α on PoLoBag performance for dataset I.

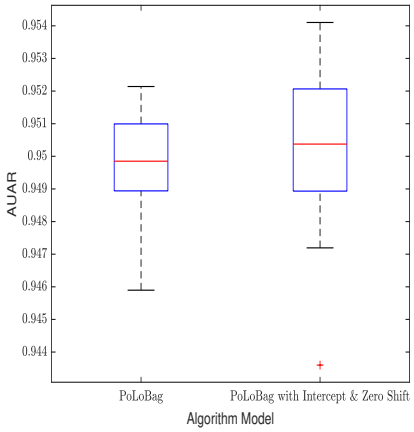
Figure 3.6: Effect of parameters on PoLoBag performance for dataset I.

3.3.3 Use of Data Shift and No Intercept Model

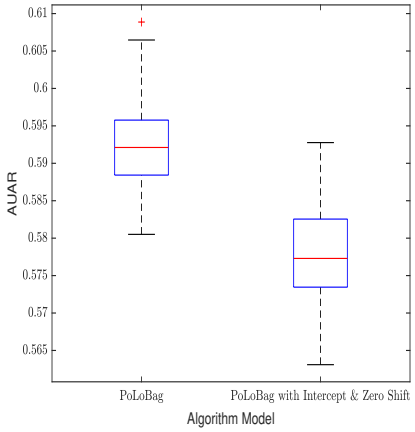
In this section we compare the performance of PoLoBag algorithm with the version of the PoLoBag algorithm where no pre-processing data shift was performed ($\tau = 0$) and instead there was an intercept term in the Lasso model. The parameter settings are same in both cases, where $n_2^1 = 0.5, n_2^2 = 3.5, n_M = 0.5, n_B = 500, \alpha = 0.1$. The results of 30 independent runs for four test datasets are shown in box plots.



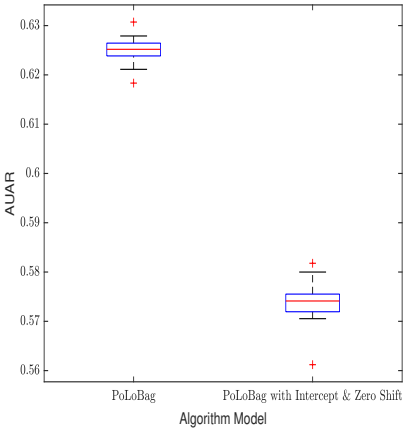
(a) Effect of data shift and no intercept on PoLoBag performance for dataset G.



(b) Effect of data shift and no intercept on PoLoBag performance for dataset A.



(c) Effect of data shift and no intercept on PoLoBag performance for dataset H.



(d) Effect of data shift and no intercept on PoLoBag performance for dataset I.

Figure 3.7: Effect of data shift and no intercept on PoLoBag performance

3.3.4 Parameter Settings for Banjo and SIREN Algorithms

In this section we present the parameter values used in Banjo and SIREN algorithms.

Parameter	A	B	C	D	E	F	G	H	I
searcherChoice	SimAnneal	" -	" -	" -	" -	– " –	" -	" -	" -
proposerChoice	RandomLocalMove	" -	" -	" -	" -	– " –	" -	" -	" -
evaluatorChoice	default	" -	" -	" -	" -	– " –	" -	" -	" -
deciderChoice	default	" -	" -	" -	" -	– " –	" -	" -	" -
discretizationPolicy	q4	q4	q4	q4	q4	q3	q4	q4	q4
minMarkovLag	0	" -	" -	" -	" -	– " –	" -	" -	" -
maxMarkovLag	0	" -	" -	" -	" -	– " –	" -	" -	" -
equivalentSampleSize	1.0	" -	" -	" -	" -	– " –	" -	" -	" -
maxParentCount	8	8	8	8	8	10	10	10	10
defaultMaxParentCount	10	" -	" -	" -	" -	– " –	" -	" -	" -
maxTime	30 m	45 m	60 m	60 m	80 m	180 m	60 m	15 m	45 m
maxRestarts	10000	" -	" -	" -	" -	– " –	" -	" -	" -
minNetworksBeforeChecking	1000	" -	" -	" -	" -	– " –	" -	" -	" -
nBestNetworks	1	" -	" -	" -	" -	– " –	" -	" -	" -
screenReportingInterval	20 s	" -	" -	" -	" -	– " –	" -	" -	" -
fileReportingInterval	10 m	" -	" -	" -	" -	– " –	" -	" -	" -
initialTemperature	10000	" -	" -	" -	" -	– " –	" -	" -	" -
coolingFactor	0.7	" -	" -	" -	" -	– " –	" -	" -	" -
reannealingTemperature	800	" -	" -	" -	" -	– " –	" -	" -	" -
maxAcceptedNetworksBeforeCooling	2500	" -	" -	" -	" -	– " –	" -	" -	" -
maxProposedNetworksBeforeCooling	10000	" -	" -	" -	" -	– " –	" -	" -	" -
minAcceptedNetworksBeforeReannealing	500	" -	" -	" -	" -	– " –	" -	" -	" -
precomputeLogGamma	yes	" -	" -	" -	" -	– " –	" -	" -	" -
useCache	fL2	fL2	fL2	fL2	fL2	fL1	fL2	fL2	fL2
cycleCheckingMethod	dfs	" -	" -	" -	" -	– " –	" -	" -	" -

Table 3.3: Banjo parameters used for experimental datasets. fL1 - fastLevel1, fl2 - fastLevel2.

Parameter	A	B	C	D	E	F	G	H	I
Number of bins	10	"	"	"	"	"	"	"	"
Spline order	2	"	"	"	"	"	"	"	"
Scoring function	S_1	"	"	"	"	"	"	"	"
Rescaling matrix	M_3	"	"	"	"	"	"	"	"

Table 3.4: SIREN parameters used for experimental datasets.

3.3.5 Effect of Dimensionality on PoLoBag Performance

The inference accuracy of PoLoBag is dependent on the number of measurement conditions in the expression dataset. This is demonstrated here by comparing PoLoBag's performance on simulated dataset C consisting of 500 measurement conditions, with those on the same dataset using 25%, 50% and 75% of the measurement conditions. For each set of conditions, the AUAR values from 30 independent PoLoBag runs are presented in box plots.

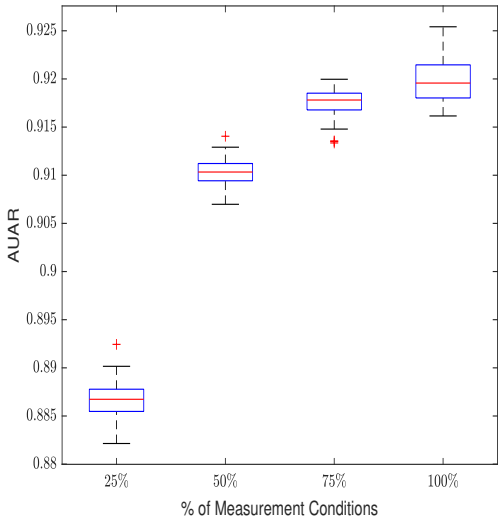
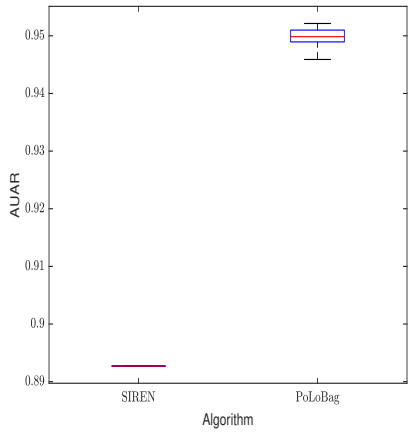


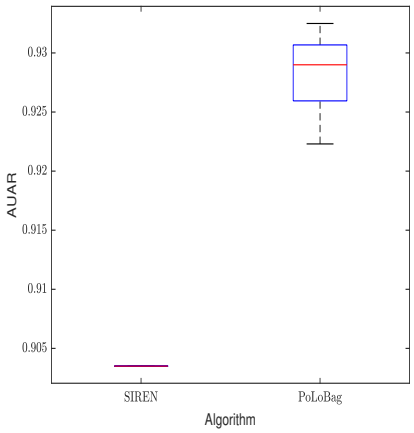
Figure 3.8: Effect of dimensionality on performance of PoLoBag for dataset C.

3.3.6 Statistical Performance Comparison of PoLoBag with Banjo and SIREN

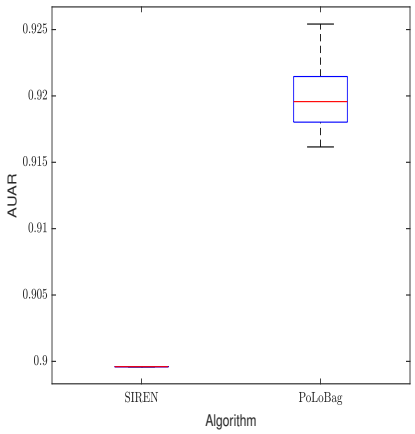
The results of PoLoBag are compared with that of Banjo and SIREN in all the test datasets. A Wilcoxon signed rank test (Wilcoxon, 1992) with the AUAR values from 30 independent runs of PoLoBag and Banjo gave a p-value of 1.7344×10^{-06} in all the test datasets. So we are able to reject the null hypothesis at 5% significance level and conclude that the difference in the results obtained from both algorithms is significant. The AUAR values from the 30 independent runs for PoLoBag are presented in a box plot along with just the AUAR value obtained from the single SIREN run for each dataset. These experiments demonstrate the significance of the difference between signed inference performance of PoLoBag and those of Banjo and SIREN.



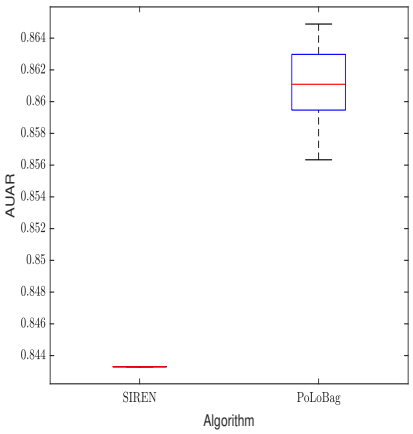
(a) Performance comparison of PoLoBag with SIREN for dataset A.



(b) Performance comparison of PoLoBag with SIREN for dataset B.

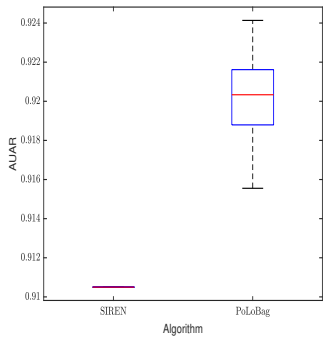


(c) Performance comparison of PoLoBag with SIREN for dataset C.

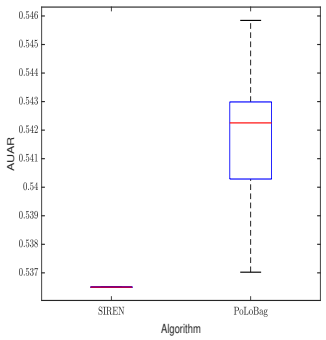


(d) Performance comparison of PoLoBag with SIREN for dataset D.

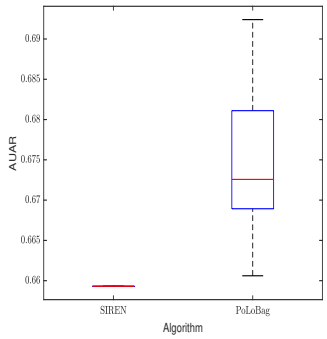
Figure 3.9: Performance comparison of PoLoBag with SIREN for datasets A-D.



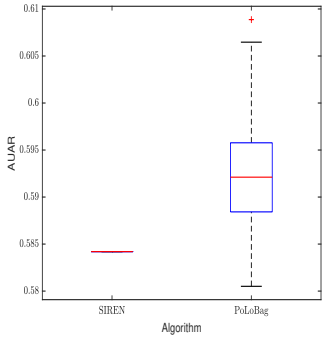
(a) Performance comparison of PoLoBag with SIREN for dataset E.



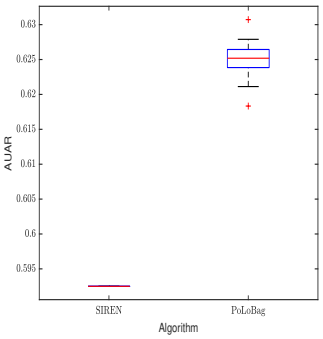
(b) Performance comparison of PoLoBag with SIREN for dataset F.



(c) Performance comparison of PoLoBag with SIREN for dataset G.



(d) Performance comparison of PoLoBag with SIREN for dataset H.



(e) Performance comparison of PoLoBag with SIREN for dataset I.

Figure 3.10: Performance comparison of PoLoBag with SIREN for datasets E-I.

3.3.7 Unsigned Performance Comparison

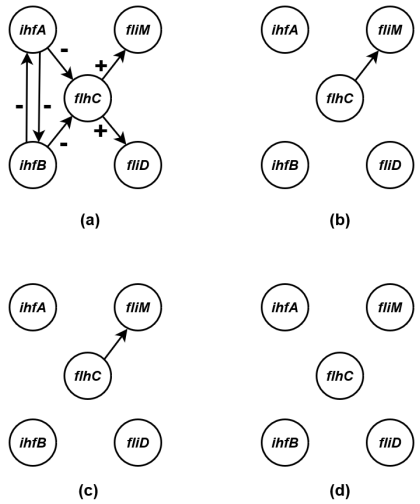
In this section the performance of PoLoBag is compared with that of GENIE3 algorithm. Since GENIE3 can not predict edge signs, here the unsigned directed performance comparison is done on the test datasets with directed networks. For the real datasets H and I, the networks consist primarily of edges having positive or negative signs defined with high confidence in the ground truth. The default parameter values in the R/bioconductor package GENIE3 (Huynh-Thu, Irrthum, et al., 2010; Aibar et al., 2017) were used. GeneNetWeaver evaluation was performed where the area under the ROC curve (AUROC) and the area under the precision-recall curve (AUPR) were computed, as in the DREAM4 challenge. Both PoLoBag and GENIE3 are non-deterministic, so the mean and standard deviation from 30 independent runs are presented.

Dataset	GENIE3 AUPR	PoLoBag AUPR	GENIE3 AUROC	PoLoBag AUROC
A	**0.1332±0.0023**	0.0715±0.0015	**0.7971±0.0016**	0.6971±0.0044
B	**0.1091±0.0008**	0.0846±0.0006	**0.7710±0.0018**	0.6987±0.0053
C	**0.1272±0.0005**	0.0955±0.0006	**0.7644±0.0017**	0.6831±0.0031
D	**0.1066±0.0007**	0.0788±0.0005	**0.7866±0.0014**	0.7174±0.0037
E	**0.0876±0.0005**	0.0662±0.0003	**0.8112±0.0017**	0.7004±0.0034
H	0.0259±0.0005	**0.0315±0.0007**	0.5151±0.0034	**0.5871±0.0055**
I	0.0135±0.0001	**0.0140±0.0001**	0.5134±0.0029	**0.5167±0.0028**

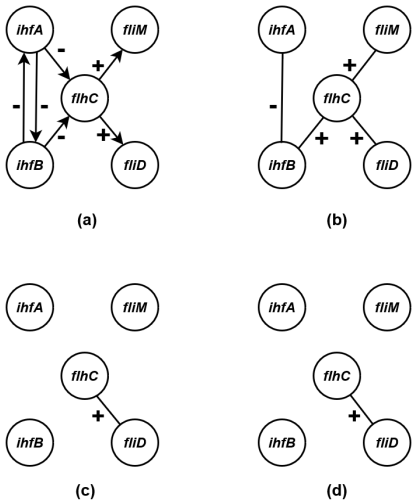
Table 3.5: Unsigned directed performance comparison between PoLoBag and GENIE3. The mean and standard deviation from 30 independent runs are given.

3.3.8 Comparison of Inferred Networks

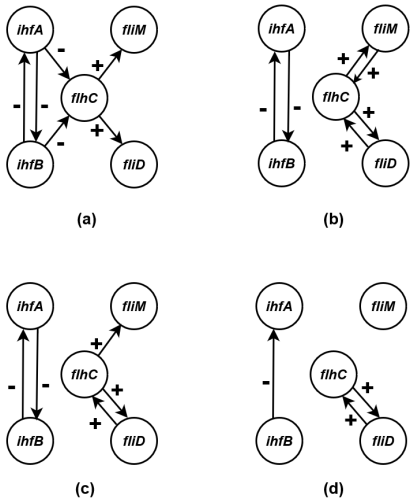
We present selected small subnetworks from the networks inferred by Banjo, SIREN and PoLoBag for two test datasets. Threshold values were used to infer networks from the algorithm output. For Banjo, we chose the threshold to be a cut-off such that out of the 30 Banjo runs, an edge has to be inferred, even with a zero influence score, in more than those many number of runs. The sign of the edge is obtained from the average score over 30 runs. For SIREN, the threshold is applied to the absolute value of the SIREN scores from the single run. One threshold value used was the best one suggested by the authors of SIREN. For PoLoBag, the edge weights are first normalized with respect to the highest absolute weight value in each run and the average absolute edge weights from 30 runs are computed. The cut-off threshold as an average relative fraction of the highest magnitude weight is applied. For each algorithm, the inferred subnetworks for a set of three threshold values are shown.



(a) Banjo inferred subnetworks for dataset E. (a) Ground truth. (b) Threshold = 1. (c) Threshold = 3. (d) Threshold = 4.

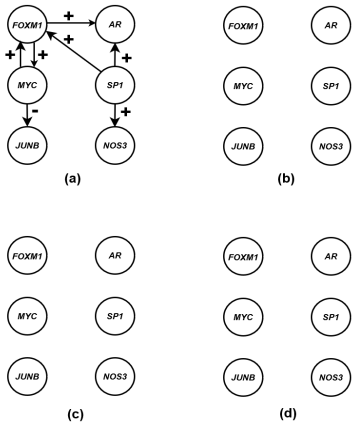


(b) SIREN inferred subnetworks for dataset E. (a) Ground truth. (b) Threshold = 0.05. (c) Threshold = 0.1. (d) Threshold = 0.158.

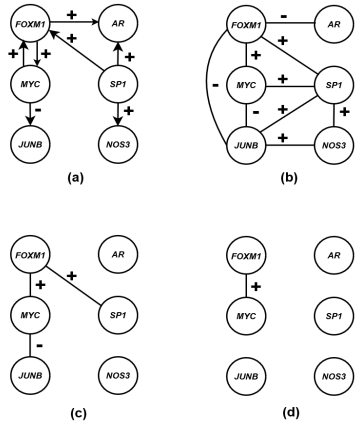


(c) PoLoBag inferred subnetworks for dataset E. (a) Ground truth. (b) Threshold = 0.05. (c) Threshold = 0.1. (d) Threshold = 0.15.

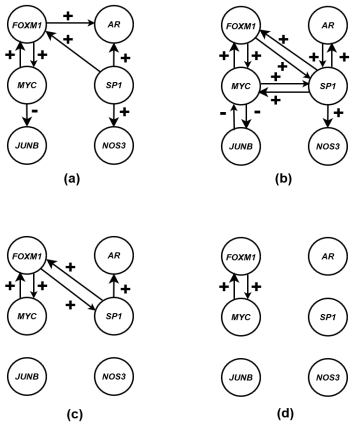
Figure 3.11: Inferred subnetworks for dataset E.



(a) Banjo inferred subnetworks for dataset I. (a) Ground truth. (b) Threshold = 1. (c) Threshold = 3. (d) Threshold = 4.



(b) SIREN inferred subnetworks for dataset I. (a) Ground truth. (b) Threshold = 0.05. (c) Threshold = 0.1. (d) Threshold = 0.158.



(c) PoLoBag inferred subnetworks for dataset I. (a) Ground truth. (b) Threshold = 0.05. (c) Threshold = 0.1. (d) Threshold = 0.15.

Figure 3.12: Inferred subnetworks for dataset I.

3.4 Discussion

In this work we have proposed the PoLoBag algorithm for signed gene regulatory network inference. It is an ensemble regression technique where the Lasso weight estimates from many bootstrap samples with polynomial features are combined in a bagging framework. Experiments on simulated and real-world datasets show that our algorithm infers the network signs more accurately than existing signed inference algorithms. It overcomes the drawbacks of Banjo and SIREN algorithms as the inferred networks have edge directions and can have cycles. Inferring the edge signs more accurately helps in enhanced analysis of the network dynamics, which is key to understanding the cellular decision-making in several important biological processes. An improved knowledge of regulatory network edge signs can provide improved understanding of the role of these networks in disease progression. This can subsequently lead to better identification of drug targets and more informed analysis of drug mode of action, with significant overall impacts on drug design or repurposing.

Future work implementing a parallel version of PoLoBag algorithm could effectively reduce run time. For dataset G, PoLoBag's total execution time was approximately 15 minutes on an i5-4590 3.30 GHz CPU with 8 GB of RAM. SIREN on Cytoscape including input-output needed around half of a minute. However, the scope of parallelization that already exists in the PoLoBag algorithm has not yet been utilized. There is a separate regression problem for each target gene, which can be run in parallel. Also in the ensemble each Lasso works on a different bootstrap sample. Another direction of possible future work can be modifying PoLoBag to additionally handle dynamic data. Currently, we do not consider any relationship in time between measurement points. Even if there are some time point measurements like in dataset F, PoLoBag considers them as separate steady-state measurements. Ablity to infer from both steady-state and time-series data can provide further insights into regulatory networks.

3.5 Combining signed architecture and dynamical information for emergent state prediction

Next we aim to investigate the role of signed GRN architecture and dynamical information in predicting the emergent states of the gene regulatory system under different conditions. Here our objective is finding top regulators which can be more important than other nodes in how they control the network emergent behavior.

3.5.1 Background and related work

In this section we discuss the study in Narang et al., 2015, which we build our work on. In Narang et al., 2015 the general human network is first constructed from transcription factor and microRNA target data collected from public databases. The general network is not complete, but based on the analysis of evolution of network properties with addition of new information the authors argue that the observed trends with this partial network can be extrapolated to the whole network. Next, the subnetwork of genes from the general network that responds to treatment with estrogen in the MCF-7 breast cancer cell line is the focus of their study. Different network analysis algorithms have been subsequently used to rank this MCF-7 estrogen response subnetwork's regulatory genes – transcription factors with at least one target in the subnetwork and microRNAs. The ranking methods used are as follows:

1. Most differentially expressed regulators – The regulators which are most differentially expressed in the MCF-7 cell line on treatment with estrogen are ranked at the top.

2. Maximum out-degree – The regulators are ranked based on the number of outgoing edges in the MCF-7 ER subnetwork.

3. Maximum target fold enrichment – First the ratio of the number of target genes of a regulator and the total number of genes are computed separately for the MCF-7 subnetwork and the general network. The regulators are ranked on the basis of the ratio of these two values.

4. Maximum in-degree – The regulators are ranked based on the number of incoming edges in the MCF-7 ER subnetwork.

5. Maximum closeness centrality – Closeness centrality (Freeman, Roeder, and Mulholland, 1979) of a regulator is the reciprocal of the sum of the shortest path distances from the regulator to every other node in the MCF-7 ER subnetwork.

6. Maximum betweenness centrality – Betweenness centrality (Brandes, 2001; Brandes, 2008) of a regulator is the sum of the fraction of all shortest paths between two nodes in the MCF-7 ER subnetwork that pass through the regulator.

7. Maximum pagerank – PageRank is an algorithm proposed to rank web pages (Page et al., 1999). The regulators are ranked on the basis of the structure of the incoming edges.

8. Innermost K-core

The K-core of a network is a maximum subnetwork where the degree of every node is greater than or equal to K (Seidman, 1983). The degree of a node here is the sum of in-degree and out-degree, which measure the number of edges coming in and going out of the node respectively. The core number of a node is the highest order K of the core that node is a part of. The K-core decomposition algorithm (Batagelj and Zaversnik, 2003) works by iteratively removing nodes that have degree less than K, along with associated edges, until a final irreducible innermost core is left. For innermost K-core ranking, the authors use nodes

in the inner cores of the network. The inner cores contain nodes that have the greatest potential for information spread within a complex network (Kitsak et al., 2010).

The regulator rankings from the methods mentioned above are validated in three ways:

1. Randomization test - This is to test if the top regulators are selected because of characteristics of the general human network or the specific MCF-7 estrogen response subnetwork. These ranking methods are applied to 10,000 randomly sampled subnetworks with similar proportion of node types as in the MCF-7 ER subnetwork. The coefficient of determination between the average regulator rank in a randomly sampled subnetwork and the regulator rank in the MCF-7 ER subnetwork is used as a measure.

2. Literature validation - The biological relevance of the top regulatory genes identified by the ranking methods is assessed based on Google Scholar citations.

3. Gene expression modeling - The top regulatory genes are used to predict the expression levels of other genes in the network. The gene expression levels are categorized into binary states of up or down as done in Middendorf et al., 2004; Natarajan et al., 2012. These states are the output variable in the classification problem.

The results show that ranking strategies of most differentially expressed and maximum in-degree criteria have poor literature validation i.e. lowest relevance of the top ranked regulators based on the number of associated publications, and have poor gene expression modeling i.e. lowest state classification accuracy, while those of maximum out-degree and closeness centrality have poor randomization metrices i.e. ranking very similar in MCF-7 ER subnetwork and randomly sampled subnetworks. The methods of betweenness centrality, pagerank and innermost K-core fare well in all 3 evaluation criteria. Out of these three ranking methods, the best literature evidence scores are obtained by betweenness centrality

and pagerank, whereas the top ranked regulators identified by innermost K-core best model the gene expression of other genes.

The K-core algorithm is used to arrange the regulatory nodes in a layered hierarchy. Similar work has been done for Yeast and E. coli in (Balcan et al., 2007; Malkoç, Balcan, and Erzan, 2010). The nodes in the inner cores are found to be more predictive of gene expression and more biologically relevant based on the validation techniques discussed above.

3.5.2 Datasets and networks

In this work we use already available GRN architecture knowledge. The signed GRN architectures and gene expression datasets used in our experiments are presented in Table 3.6. The simulated data used is multifactorial data, which are steady-state measurements obtained from applying multifactorial perturbations to the underlying network. These perturbations are simulated by slightly increasing or decreasing the basal activation of all network nodes by random amounts. These were generated using the tool GeneNetWeaver or GNW (Schaffter, Marbach, and Floreano, 2011; Marbach, Schaffter, Mattiussi, et al., 2009). In this tool network structures are obtained by extracting modules from known biological networks of organisms like Yeast (*S. cerevisiae*) and *E. coli*. Then these network topologies are given dynamical models of gene regulation. Both independent and synergistic regulatory interactions are taken into account. Molecular noise and measurement noise are modeled into the system. The GNW data generation settings were those used in DREAM4 *In Silico Multifactorial* subchallenge – coefficient of the molecular noise in the stochastic simulation = 0.05, the measurement noise as a mix of normal (standard deviation = 0.025) and lognormal (standard deviation = 0.075) noise, and normalization after adding the measurement noise.

In our experiments we also used real biological data. The first real dataset is the ex-

Dataset	Organism	Type	Total Nodes	Regulators	Edges
Ecoli	*E. coli*	Simulated	1565	176	3648
Yeast	Yeast	Simulated	2500	153	10528
Ecoli-1	*E. coli*	Real	1409	161	2310
Ecoli-2	*E. coli*	Real	1456	172	2424
Human	Human	Real	1665	519	3954

Table 3.6: Experimental datasets for emergent state prediction. The regulators are nodes with outgoing edges in these networks.

pression for *E. coli* genes obtained from the Many Microbe Microarray Database M^{3D} (Faith et al., 2007). The second real dataset was collected from the GEO database with accession number GSE135516, which was obtained from growth of *E. coli* with various supplements (Anand et al., 2020). The underlying ground truth regulatory network for both datasets was obtained from the RegulonDB database (Gama-Castro et al., 2008). The third real dataset used in our experiments is for human genes collected from the GEO database with accession number GDS2545. The measurement samples correspond to gene expression measurements in four states - normal prostate tissue, normal prostate tissue adjacent to tumor, primary prostate tumor tissue and metastatic prostate cancer tissue (Chandran et al., 2007). The human gene regulatory network was obtained from the manually curated TRRUST database (H. Han et al., 2018).

3.5.3 Algorithm

The objective of this work is to demonstrate that combination of signed architecture and dynamical information leads to better identification of top regulators in a GRN and hence improved prediction of network emergent states. For each measurement sample referring to

a measurement condition, we consider a part of the overall network given in Table 3.6 to be active, consisting of nodes which are either up or down regulated. This can be determined based on the wild type measurement values of the genes, and a fold expression change above or below a certain threshold can signify the gene to be an up or down state respectively. However the wild type measurements are not always available. So we first compute the z-scores of the genes expression values over all available measurement samples. For each measurement sample, these z-score values are sorted. The genes with the highest 200 values are considered to be in the up state, and those with the lowest 200 values are considered to be in the down state. This is similar to the approach used for gene state classification in different cell lines (Natarajan et al., 2012).

Now for each active 400 node subnetwork for a measurement sample, we identify the top regulators using these three methods.

1. K-core - We use K-core decomposition on the active subnetwork. All the regulator nodes in the innermost core, that is with the highest core number, are considered to be the top regulators. If there are disconnected components with the highest core number, we consider the nodes in all the components. One key difference with the work in Narang et al., 2015 is that the authors apply K-core decomposition on the subnetwork of regulators within the active subnetwork, however here we apply this decomposition on the entire active subnetwork for improved subsequent top regulator selection. Another important difference is that we use regulator nodes only from the innermost core, and not from multiple inner cores.

2. Dynamical K-core - This proposed method combines signed architecture and dynamical information, before application of K-core decomposition method. Top regulator nodes would be potentially central to the information flow, and with this motivation we want to consider the information flow between the regulator nodes under a particular

measurement condition.

Let the state of a node i in measurement sample k be denoted by x_k^i where $x_k^i \in \{-1, +1\}$, as we consider the active subnetwork to be comprised of nodes only in up or down state. For an edge e^{ij} between regulator nodes i and j we consider a successful flow of information in measurement condition k when the following criterion is met, given by

$$x_k^i * sgn(e^{ij}) = x_k^j. \tag{3.11}$$

The sgn is the sign function which denotes the sign of the edge, either +1 or −1 (the value of 0 denotes a non existent edge). We delete the edges in the subnetwork where this criterion is not met and then apply the k-core decomposition to identify the innermost core nodes or top nodes. We refer to this as the dynamical K-core method. A point to note is that as we apply this to the network target node state classification problem as described in the following section, where the states of the regulators are assumed to be known and the states of the target genes are to be predicted, and therefore this edge deletion is done only for edges between regulators .

3. Random - Top regulators are selected randomly from the set of regulators in the active subnetwork.

3.5.4 Validation

For validation we use the gene expression modeling used in Narang et al., 2015. The idea is that top regulator nodes which are more important would be able to better predict the states of the target nodes in the network. Let there be m top regulators identified by a ranking method. The total number of target genes (which are not regulators themselves) is n. A target gene $j \in \{1, \cdots, n\}$ is represented as (X_k^j, y_k^j) for measurement condition k. The label $y_k^j = x_k^j$ denotes the state of the node j, either +1 or −1 in the active subnetwork. The feature

vector X_k^j is a m dimensional vector, where each element is of the form $x_k^i * sgn(e^{ij})$ if an edge e^{ij} exists between top regulator $i \in \{1, \cdots, m\}$ in state x_k^i and target node j, otherwise is 0. This is slightly different from the representation used in Narang et al., 2015, where for an existing edge between top regulator i and target gene j, the corresponding element of X_k^j would be 1. With the modified target gene representation as in this work, we obtained better performance.

We solve this state prediction problem using support vector machine algorithm, which gives the highest classification performance in the experiments in Narang et al., 2015. The optimal set of algorithm parameters were obtained using a grid search over a range of values, which was the kept the same as the previous work. The average AUROC obtained from 5-fold cross validation is used as the performance metric here. However in the previous work, the state prediction is done only for one measurement condition, that is for MCF-7 treatment with estrogen. Here we looked at state prediction problems in each of the measurement samples separately, and the mean and the standard deviation of the AUROC metrics over the samples are presented. This can provide an understanding of how well a method identifies top regulators over many measurement conditions.

3.5.5 Results

In this section we present the results of comparison of K-core and dynamical K-core on different test datasets. Firstly the number of top or innermost core regulators identified by both methods are presented in Table 3.7. The values shown in the table are averaged over all the measurement samples. As mentioned, here the K-core decomposition is applied to the entire 400 node network.

In this work we are primarily interested in the performance difference between K-core

Dataset	Total Measurement Samples	Average number of innermost core regulators from K-core	Average number of innermost core regulators from dynamical K-core
Ecoli	200	7.88	8.53
Yeast	200	11.13	10.83
Ecoli-1	179	7.13	7.43
Ecoli-2	24	7.58	7.75
Human	171	13.68	13.67

Table 3.7: Number of regulators in the innermost core obtained from K-core and dynamical K-core methods averaged over all measurement samples.

and dynamical K-core. So going forward the results are presented for relevant measurement samples, which are the ones out of the total measurement samples in which the innermost core regulators identified by K-core and dynamical K-core are different. The results are presented in Table 3.8 where we show the mean and the standard deviation obtained from four methods. Apart from K-core and dynamical K-core, we also have the random method. For random selection of top nodes, the selection for one measurement sample would be repeated many number of times to obtain an average metric. However, here for one measurement sample we did the selection only once, and since the mean is shown over many samples where random regulators are selected independently, the overall performance of the random selection method is depicted.

In Table 3.8 we also present the performance of maximum out-degree regulator ranking. For maximum out-degree, the number of top regulators used was the same as the number of innermost core regulators in dynamical K-core. In Narang et al., 2015 the performance of maximum out-degree and maximum closeness centrality rankings in terms of the gene expression modeling is found to be close to that of innermost K-core. In this work we select

Dataset	Relevant Measurement Samples	Random	K-core	Dynamical K-core	Maximum out-degree
Ecoli	78	0.5567± 0.0562	0.6283± 0.0806	0.6438± 0.0766	**0.6661± 0.0783**
Yeast	94	0.6139± 0.0583	0.6589± 0.0582	0.6575± 0.0574	**0.6714± 0.0583**
Ecoli-1	78	0.5392± 0.0444	0.5718± 0.0548	0.5976± 0.0609	**0.6424± 0.0653**
Ecoli-2	12	0.5257± 0.0171	0.5778± 0.0333	0.5827± 0.0218	**0.5866± 0.0320**
Human	164	0.5152± 0.0177	0.5510± 0.0327	0.5513± 0.0315	**0.5522± 0.0326**

Table 3.8: Performance comparison in terms of mean and standard deviation of AUROC metrics over relevant measurement samples. Relevant measurement samples are those in which the innermost core regulators are different between K-core and dynamical K-core methods. The mean best, or the one with the highest mean, is marked in boldface.

maximum out-degree as a comparison reference.

The results show that in 4 out of the 5 test datasets, the mean classification performance of dynamical K-core is better than that of K-core. Individually in some measurement samples, K-core gives better performance than dynamical K-core, however the mean over a number of relevant measurement samples points to the usefulness of the top nodes identified by dynamical K-core in predicting emergent network state. Though dynamical K-core gives better AUROC metrics than maximum out-degree in some measurement samples, the mean metrics of maximum out-degree ranking are better than those of dynamical K-core in all the

datasets.

The higher mean emergent state prediction performance of dynamical K-core compared to K-core can not be attributed only to the larger number of nodes in the innermost core. Firstly a higher number of nodes in the innermost core does not always lead to better performance, as for the Human dataset. Secondly the difference in the average number of nodes in the innermost core for dataset Ecoli-1 is not high enough to alone account for the large observed difference in the mean performance. We observed in our experiments that increasing the number of top regulators for a measurement sample does not always lead to better performance, so the top nodes themselves rather than just the number of nodes appears to be a decisive factor.

3.5.6 Discussion

An extension of the K-core decomposition for signed architectures has been previously proposed (Giatsidis et al., 2014), where the node degrees are separately defined for positive and negative edges. Here we consider one overall degree value for a node. Also, the decomposition in Giatsidis et al., 2014 does not use the criterion of the flow of information as in our proposed dynamical K-core method.

Our results suggest that there is a trend towards dynamical K-core being better than K-core in the identification of important regulator nodes leading to better emergent state prediction. In most of our test datasets, dynamical K-core mean performance is better than that of K-core, however, in some individual measurement samples, K-core performance is better. This can potentially be the result of two factors. Firstly, for real datasets in our experiments, the networks obtained from databases might have missing or false edges. Secondly, the choice of a threshold for determining which part of the network is active in a

measurement condition is critical.

Our results also point to a trend towards maximum out-degree ranking being better than both dynamical K-core and K-core in emergent state prediction. We tried incorporating dynamical information to maximum out-degree ranking, in the form of the information flow criterion as in dynamical K-core, however with no substantial performance change. The better mean performance of the out-degree ranking method suggests the usefulness of a more complete picture of how an individual node is connected in the network in its representation. We must acknowledge that in all the tested methods, we are inherently adding dynamical information when we select the top regulators from only the active subnetworks within the networks. Still we can see the benefit in combining signed architecture with additional dynamical information in identifying top regulators and predicting emergent behavior, as suggested by the improved mean performance metrics of dynamical K-core compared to those of K-core. The effectiveness of our proposed method using signed GRN architecture in emergent state prediction can be explored in cases when a measurement condition refers to a type of cancer for example, with an active subnetwork for the cancer type obtained from prior knowledge or experiments, and the unknown emergent state needs to be predicted for a patient with that type of cancer and a particular subtype or stage.

3.6 Chapter Summary

In this chapter we first present the work in our paper (Ghosh Roy, Geard, et al., 2020), where a novel and more accurate signed GRN inference algorithm PoLoBag is proposed. The objective is to infer the architecture of the gene regulatory network from a general form of gene expression data, and the inferred architecture must have edge signs denoting activating or inhibitory regulatory relationships. Many standard GRN inference algorithms

do not produce edge signs, and algorithms which can perform such signed inference suffer from limitations like no feedback loops (Banjo), no edge directions (SIREN) and variable selection not being accurate in the presence of synergistic regulation (Lasso-based methods). In our proposed PoLoBag algorithm, we combine individual Lasso models in a bagging ensemble approach, with both data sampling and feature bagging, and each Lasso model consists of both linear and nonlinear polynomial features. We demonstrate how PoLoBag consistently gives more accurate signed inference than Banjo and SIREN algorithms on simulated and real-world datasets in Table 3.2, and how it overcomes their shortcomings by having cycles and edge directions in the inferred networks in Figure 3.2.

Second, we further use signed GRN architecture combining it with dynamical information in our proposed dynamical K-core method. We use dynamical K-core to find top regulators in the GRN, and then predict the emergent states of the network, that is the states of the target nodes when the states of the regulator nodes are known, under different measurement conditions. From Table 3.8, we observe a trend towards dynamical K-core identifying top regulators which can predict the network emergent states better than random selection and K-core methods. However, the best mean prediction performance metrics are obtained by maximum out-degree ranking. This points to the importance of analyzing where a node lies in the network, in terms of its connectivity to its neighbors and in the global architectural organization of the network.

Chapter Four

Bow-tie Architecture of Gene Regulatory Networks in Species of Varying Complexity

In this chapter we aim to investigate the existence of a global architectural feature in GRNs to understand species differences in terms of a universally present emergent property of their gene regulatory systems. A network architectural feature associated with controlling system-level dynamical properties is the bow-tie, identified by a strongly connected subnetwork, the CORE layer, between two sets of nodes, the IN and the OUT layers. Though a bow-tie architecture has been observed in many networks, its existence has not been extensively investigated in GRNs of species of widely varying biological complexity. We analyze publicly available GRNs of several well-studied species from prokaryotes to unicellular eukaryotes to multicellular organisms, and based on the results of our analysis, we aim to predict trends in the emergence of a dynamical gene regulatory system property with varying biological complexity.

The chapter is organized in the following order. Section 4.1 introduces the background

and presents related studies. Section 4.2 discusses how the GRNs are extracted and arranged in order of complexity and describes the bow-tie architecture decomposition method. Section 4.3 presents the observations of our study. In Section 4.4 we summarize our observations and from these observations deduce their biological implications. Lastly, the strengths, weaknesses and future directions of this work are discussed in Section 4.5. A brief summary of the entire chapter is provided in Section 4.6.

4.1 Background and related work

A key objective of comparative biology is explaining biological differences between species. Gene regulation plays a critical role in explaining such organismal differences (King and Wilson, 1975). Gene regulatory networks (GRNs) (Bolouri, 2008) are networks where edges connect regulator nodes, such as transcription factors (TFs), to target nodes. A GRN is a model of the gene regulatory system that controls the development, function and pathology of organisms, and hence its analysis is extremely important. Study of GRN structure and how it varies between species can provide insights into how changes in gene expression, underlying divergence in phenotypes, occur between species (Wittkopp, 2007). Differences in GRN architectural organization are considered the reason for differential dynamic regulatory behavior between eukaryotic Yeast (*Saccharomyces cerevisiae*) and prokaryotic bacteria (Rodriguez-Caso, Corominas-Murtra, and Solé, 2009). Comparison across multiple eukaryotes reveals a common architectural feature of the GRN – a scale-free topology, but with species-specific characteristics likely to produce species-specific phenotypes (Ouma, Pogacar, and Grotewold, 2018). So it is vital to analyze the differences in GRN architecture to understand differences between species.

Differences between species are exhibited at various levels like anatomy, physiology

and behavior. One approach of understanding differences between species is looking at differences in universally present dynamical regulatory system properties. Complex biological systems display some inherent system-level dynamical properties. Understanding the emergence of these properties is important for understanding the functioning and pathology of organisms. We want to investigate how the dynamical system property of controllability, ubiquitous in the context of gene regulation, has evolved differently between different species. For this purpose, analysing the architecture of their GRNs becomes crucial.

A network architecture associated with important dynamical properties like robustness, flexibility, evolvability and controllability (Csete and Doyle, 2004) is the *bow-tie*. The bow-tie architecture has been observed in various network types, including information networks (Broder et al., 2011), internet protocol networks (Akhshabi and Dovrolis, 2011), neural networks (Hinton and Salakhutdinov, 2006) and biological networks like metabolic (H.-W. Ma and Zeng, 2003) and signaling networks (Supper et al., 2009). The formal definition of the bow-tie architecture in a directed graph is given in terms of a strongly connected component (SCC) (R. Yang, Zhuhadar, and Nasraoui, 2011). A SCC is a subnetwork in which every node is connected to every other node. The largest of these, the largest strong component (LSC) in the network is defined to be the bow-tie CORE layer (Broder et al., 2011; H.-W. Ma and Zeng, 2003). The LSC CORE lies between the IN layer and the OUT layer. As presented in Figure 4.1, the rest of the nodes in the network are categorized into remaining layers of the bow-tie – INTENDRILS, OUTTENDRILS, TUBES and OTHERS.

In literature, the bow-tie architecture is also associated with an hourglass shape where the CORE is smaller than the IN and the OUT layers (Tieri et al., 2010; Friedlander et al., 2015). In this study, we use the definition of the bow-tie architecture in terms of a strongly connected component CORE between the IN layer and the OUT layer, where the hourglass shape is not compulsory. Researchers have previously shown the existence of a bow-tie architecture in GRNs of some eukaryotes, with the LSC CORE being the only non-trivial

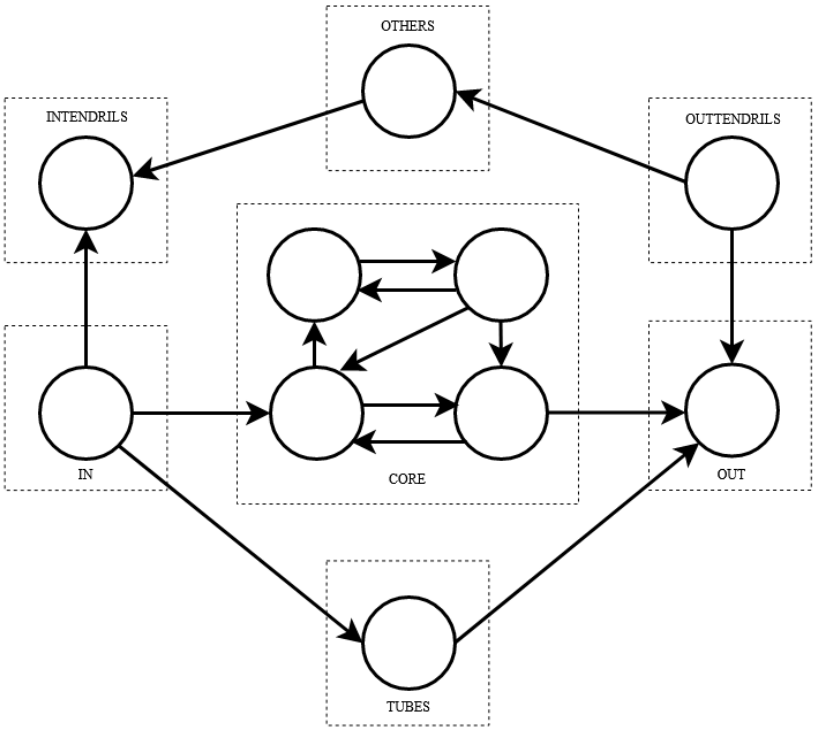


Figure 4.1: An example of a bow-tie architecture with the largest strong component (LSC) CORE layer. The circles represent nodes and the arrows represent edges. The different bow-tie layers are denoted by dashed boxes.

(consisting of more than one node) strong component. For example, the work in (Rodriguez-Caso, Corominas-Murtra, and Solé, 2009) demonstrates that a bow-tie architecture with one large strongly connected CORE is observed in the Yeast (*Saccharomyces cerevisiae*) GRN's dynamical backbone, defined as a subgraph of computationally relevant dependencies. However, the authors observed a top-down hierarchy but not a bow-tie structure in the dynamical backbones of bacteria *B. subtilis* and *E. coli* GRNs. The other example is that condition-specific TF-TF regulatory networks of the plant Arabidopsis (*Arabidopsis thaliana*) in six tested experimental conditions exhibit a bow-tie architecture with one non-trivial distinct LSC CORE (S. Luo et al., 2018). The authors in (S. Luo et al., 2018) additionally speculate

that such an architecture might be prevalent in other eukaryotic species. However, the existence of bow-tie architectures and the quantification of their characteristics across GRNs in species of a wide range of biological complexity have not yet been addressed.

The bow-tie CORE size, both absolute (number of nodes or regulators) and relative (number of nodes or regulators relative to the corresponding total number in the network), is considered to be a vital aspect of the network architecture (Csermely et al., 2013), as it is related to important dynamical system properties including controllability (Csete and Doyle, 2004).

4.2 Materials and methods

4.2.1 GRN extraction

In our study we have selected some species covering a wide range of biological complexity, for which the GRNs are readily available from public data sources. These different sources for GRNs have been created and managed by curators using methodologies differing slightly or even widely. However, in our analysis we need a common ground for GRN comparison. Our objective has been to use the GRN extraction criteria that provide, for subsequent comparative analysis, the optimal ground in terms of completeness and similarity.

GRNs can capture several forms of regulatory interactions. In the extracted networks of our analysis the regulators are TF genes, where TFs can also refer to factors classified as TFs in the data, like sigma factors in prokaryotes or co-factors or chromatin remodeling factors in eukaryotes. The target genes can represent TF, microRNA, small RNA or other genes whose transcription is controlled by these regulators. Like in (Kumar et al., 2015), we have excluded the regulatory interactions where the source genes represent non-coding

RNAs like bacterial small RNAs or microRNAs. However unlike (Kumar et al., 2015), we have incorporated the interactions where the regulators are TF genes which regulate the transcription of non-coding RNA target genes. We have aimed to use the most unique gene identifiers present in the data source and extract only the regulatory interactions with valid identifiers. Where possible, a complex/operon/heteromer is to be included in the network as its individual genes. For ease of use, we have selected only the TF-target gene interactions available in the data sources, when in some sources there can be additional related information like that of TF binding sites, promoters or gene expression correlation. The GRNs in our study are assumed to be general, and not specific to any particular experimental condition or cell type.

One important aspect in extracting the GRNs is the type and reliability of evidence associated with the interactions. An interaction can be experimentally validated or computationally predicted, and the interaction can be ranked based on the reliability of the evidence. All these different data sources use their own set of criteria for defining these interaction properties, and in some cases that information is not available. Choosing the strictest possible threshold on these interaction properties could lead to incomplete information for some species, which is not suitable for a reliable analysis. In our study, for a data source we extract all interactions with any evidence irrespective of its type and reliability. Although extracting interactions without a threshold might lead to false positive edges, it eliminates the variability of analysis caused by different selections of threshold. We have excluded interactions which are categorized as indirect in the data source.

Completeness of the data is an important factor while extracting GRNs. We have addressed the issue of incompleteness of the data sources by only considering extracted GRNs with coverage of more than 50% of the species total genes. These total gene (protein+RNA) numbers for all species were obtained from the Kyoto Encyclopedia of Genes and Genomes (KEGG) Genome database (Kanehisa, Furumichi, et al., 2021). For some species there are

Species	Data source	Extraction criteria	% total genes
E. coli	RegulonDB	All TF-target gene and sigma factor-target gene interactions	54
Yeast	YTRP	All direct TF-target gene interactions with binding evidence in the shortest pathway connecting a TF-target gene pair with expression evidence	80
Arabidopsis	AtRegNet	All direct TF-target gene interactions with TF and target gene name and locus specified	57
Drosophila	DROID	All TF-target gene interactions	81
Mouse	RegNetwork	All TF-target gene interactions	73
Human	RegNetwork	All TF-target gene interactions	99

Table 4.1: GRN data sources selected for bow-tie architecture decomposition. The percentage of species total genes (protein+RNA) in the extracted GRN, rounded to a whole number, is shown.)

multiple different data sources. To finally have one data source per species in our analysis, we have used the one with the highest percentage of the total genes in the species. The data sources and the corresponding extraction criteria for GRNs of well-studied species selected for our architecture analysis are listed in Table 4.1. The extraction criteria specific to each data source are given with the percentage of species total genes (protein+RNA) in the extracted GRN (denoted as % total genes, rounded to whole numbers). We believe that these network extraction criteria give us the most optimally complete and fair ground of comparison possible across GRNs of several species from different sources.

Among the selected GRNs, *Escherichia coli* K-12 GRN was extracted from the RegulonDB database (Santos-Zavaleta et al., 2019). The GRN contains TF-target gene and sigma factor-target gene interactions curated from literature with different ranks of experi-

mental evidence, including some which are predicted. For Yeast (*Saccharomyces cerevisiae*), the Yeast Transcriptional Regulatory Pathway (YTRP) database (T.-H. Yang et al., 2014) was used, which consists of curated interactions with evidence of either TF-target gene binding or target gene expression variation on perturbation of TF, or both. We extracted the TF-target gene direct pairs with experimental binding evidence in the shortest regulatory pathway connecting a TF and a target gene with expression evidence. The *Arabidopsis thaliana* GRN consists of different ranks of direct TF-target gene interactions obtained from the *Arabidopsis thaliana* regulatory network (AtRegNet) database available on Arabidopsis Gene Regulatory Information Server (AGRIS) (Yilmaz et al., 2010). The GRN of *Drosophila melanogaster* consists of TF-target gene interactions with experimental evidence of the TF binding to the gene and regulating its transcription, or only binding evidence, obtained from the Drosophila Interactions Database (DroID) (Murali et al., 2011). The data source used for Mouse (*Mus musculus*) and Human (*Homo sapiens*) GRNs is RegNetwork (Z.-P. Liu et al., 2015). These extracted GRNs have TF-target gene interactions with different ranks of experimental or predicted evidence. These GRNs have observed percentages of the species total genes higher than the GRNs from other data sources for these two species. The source files, extraction scripts and extracted GRNs are available at https://github.com/gourabghoshroy/Bow-tieGRN.

The data sources presented in Table 4.2 were not selected because they do not have more than 50% of total genes or there is another data source for the same species with a larger % total genes. The GRNs for prokaryotes *Bacillus subtilis* 168 and *Corynebacterium glutamicum* ATCC 13032 were obtained from the database CoryneRegNet 7.0 (Parise et al., 2020) containing regulatory interactions with either experimental or predicted evidence, or both. The GRN for Rat (Rattus norvegicus) was collected from Open-access Repository of Transcriptional Interactions (ORTI) (Vafaee et al., 2016). The GRN consists of retrieved TF-target gene interactions with different ranks of evidence. ORTI consists primarily of in-

Species	Data source	Extraction criteria	% total genes
B. subtilis	CoryneRegNet	All TF-target gene interactions (sigma factor-target gene interactions included)	15
C. glutamicum	CoryneRegNet	All TF-target gene interactions (sigma factor-target gene interactions included)	30
Rat	ORTI	All TF-target gene interactions with TF and target gene id specified	4
Mouse	ORTI	All TF-target gene interactions with TF and target gene id specified	13
	TRRUST2	All TF-target gene interactions	9
Human	ORTI	All TF-target gene interactions with TF and target gene id specified	78
	TRRUST2	All TF-target gene interactions	13
	ENCODE	All TF-target gene interactions	46

Table 4.2: GRN data sources not selected for bow-tie architecture decomposition. The percentage of species total genes (protein+RNA) in the extracted GRN, rounded to a whole number, is shown. These data sources were not selected because they do not have more than 50% of total genes or there is another data source for the same species with a larger % total genes.

teractions also for Mouse (*Mus musculus*) and Human (*Homo sapiens*). For the two species Mouse and Human, GRNs were also constructed from the database Transcriptional Regulatory Relationships Unraveled by Sentence-based Text mining (TRRUST) version 2 (H. Han et al., 2018). Another source for Human GRN was obtained from the ENCODE project (M. B. Gerstein et al., 2012).

4.2.2 Characterization of species complexity

In this section we describe how we have characterized the notion of biological complexity in our analysis. The complexity of an organism can be defined in many ways, like genomic complexity (Adami, Ofria, and Collier, 2000) and phenotypic complexity (Marion, Fordyce, and Fitzpatrick, 2015). In our study, the six species for which GRNs are selected are arranged in an order of complexity defined on the basis of their number of cell types (Hedges et al., 2004). A widely accepted precise definition of a cell type is not available, and researchers have used mostly morphological characteristics to differentiate between types (Bell and Mooers, 1997). More recently, cell types form a controlled vocabulary in the Cell Ontology (Diehl et al., 2016), and cells can be classified into these types using the OnClass algorithm (Sheng Wang et al., 2021). Rather than being based on phenotypic similarity, an evolutionary definition of cell types is also available now (Arendt et al., 2016). The stable equilibrium states or gene expression patterns of GRNs are viewed to be corresponding to gene expression profiles associated with each cell type (S. Kauffman, 1969). So we believe that this definition of biological complexity is relevant in our study where we analyze GRNs of different species. As mentioned previously, the studied general GRNs are not specific to any particular cell type.

We have used the knowledge about the number of cell types of different species from literature (Hedges et al., 2004; Bell and Mooers, 1997). When the data for a particular species was not available in the used sources, we have utilized the maximum number of cell types observed in the major group the species belongs to. *E. coli* is the simplest organism in our study as it is a prokaryotic eubacteria, which have a maximum of 2 cell types. Unicellular eukaryote Yeast is ranked next in complexity with maximum 3 cell types in Saccharomyces genus. For the phyla of Arabidopsis and Drosophila, the number of maximum observed cell types are 44 and 69 respectively and hence they are arranged in that order. The next more

complex species is Mouse with 102 cell types. Finally we have the species Human with 411 cell types including 145 types of neurons (Vickaryous and Hall, 2006). We have used this order of complexity in presenting all our results.

4.2.3 Bow-tie architecture decomposition

To analyze the architecture of GRNs, we have used the strongly connected component based bow-tie architecture decomposition (R. Yang, Zhuhadar, and Nasraoui, 2011). In some other definitions, the bow-tie network structure needs to resemble an hourglass, with the intermediate CORE smaller than the input and output layers (Friedlander et al., 2015). However, this bow-tie definition, as used in our work, does not have this particular requirement. The details of the decomposition are given as follows. Let a directed network G be represented with a set V of vertices and a set E of edges. A destination node is defined to be reachable from a source node if there is a directed path from the source to the destination node. This definition of reachability (to or from) is extended to sets of nodes if there is a path to or from at least one node in that set. A strongly connected component is a set of nodes where every node is reachable from every other node in the set. By definition, every single node is a trivial strongly connected component. The bow-tie decomposition of the network $G = (V, E)$ with the largest strong component (LSC) defined to be the CORE decomposes the network (see Figure 4.1) into these seven different layers or sets of nodes:

1. CORE = LSC
2. IN = $\{ v \in V - \text{CORE} \mid \text{CORE is reachable from } v \}$
3. OUT = $\{ v \in V - \text{CORE} \mid v \text{ is reachable from CORE} \}$
4. INTENDRILS = $\{ v \in V - \text{CORE} \mid v$ is reachable from IN and OUT is not reachable from $v \}$

5. OUTTENDRILS = { $v \in V$ – CORE | v is not reachable from IN and OUT is reachable from v }

6. TUBES = { $v \in V$ – CORE – IN – OUT | v is reachable from IN and OUT is reachable from v }

7. OTHERS = V – CORE – IN – OUT – INTENDRILS – OUTTENDRILS – TUBES.

The bow-tie decomposition is performed using Algorithm 2. $DFS_G(v)$ represents the set of nodes obtained from a depth-first search starting at vertex v in network G. G^T refers to the network that is obtained by reversing the direction of every edge in G.

4.2.4 Null model construction

We compared the GRNs of different species with their randomized counterparts in which the number of nodes and the degree at each node are preserved. Similar to the approach in (Rodriguez-Caso, Corominas-Murtra, and Solé, 2009), we generate these random networks. The autoregulatory edges of the original GRN are preserved separately because they do not affect the bow-tie layer definitions. This random generation process starts with the other non-autoregulatory edges in the original GRN forming the initial edge list. A pair is selected randomly from this list and their end nodes are swapped. If any of these new edges lead to self loops or multiple edges, this swap operation is not performed for that pair. After trying the swap operation on every distinct pair in the edge list for an iteration, the algorithm in the next iteration repeats the process on the new edge list, consisting of edges from the pairs which could not be swapped. To make the process efficient on one hand and to have enough iterations for many swap operations to possibly occur on the other, we chose the number of iterations to be 10. There can be some edges whose end nodes are not swapped

Algorithm 2 Bow-tie network decomposition algorithm (R. Yang, Zhuhadar, and Nasraoui, 2011) based on the largest strong component (LSC) as CORE layer.

Set CORE = LSC.
Select a $v \in$ CORE. IN = $DFS_{G^T}(v)$ – CORE.
Select a $v \in$ CORE. OUT = $DFS_G(v)$ – CORE.
foreach $v \in V$ – CORE – IN – OUT **do**
    IRV = (IN $\cap$ $DFS_{G^T}(v) \neq \phi$).
    VRO = (OUT $\cap$ $DFS_G(v) \neq \phi$).
    if IRV and not VRO **then**
        $v \in$ INTENDRILS.
    else if not IRV and VRO **then**
        $v \in$ OUTTENDRILS.
    else if IRV and VRO **then**
        $v \in$ TUBES.
    else
        $v \in$ OTHERS.
    end if
end foreach

with another edge even after the 10 iterations. There are other ways of generating these null model networks, here we have used this simple and fast method for our analysis. 1000 such random networks were generated independently for each GRN.

4.3 Results

In this section we present the results of applying the bow-tie architecture decomposition (R. Yang, Zhuhadar, and Nasraoui, 2011) (described in Section 4.2.3) on the selected GRNs of six species of varying complexity. Table 4.3 shows the number of nodes and regulators in each of the bow-tie layers in these GRNs, where regulators are nodes with at least one outgoing edge in the extracted GRN. We present the relative sizes of these layers with respect to all nodes and all regulators in the network in Figure 4.2A and Figure 4.2B respectively.

From Table 4.3 we observe that for all these GRNs there is a non-trivial LSC substantially larger than the 2^{nd} LSC. For example in *E. coli* GRN, the LSC consists of 54 nodes compared to a 3-node 2^{nd} LSC, and the difference between the two are larger for other species. In all these GRNs, this LSC is the distinct CORE of the bow-tie, located between a smaller IN layer and a larger OUT layer. As evident from Figure 4.2B, the non-trivial CORE which consists only of regulators by definition, consists of a substantial percentage of all regulator nodes, specially for eukaryotes ($> 40\%$). We can therefore conclude that a bow-tie architecture with one distinct LSC CORE exists in the GRNs of all these species of varying complexity.

The GRN bow-tie architecture observed in our results has some important differences between species. Through the arrangement of species in an increasing order of biological complexity from *E. coli* to Human, in Table 4.3 and Figure 4.2, we observe the relationship of the bow-tie CORE size with this biological complexity. Since we are comparing differently

Layer		*E. coli*	Yeast	Arabidopsis	Drosophila	Mouse	Human
All	Edges	7348	16032	670771	157462	120579	171946
	Nodes	2381	5124	16427	12323	18916	22121
	Regs	220	159	573	149	1328	1456
CORE	Nodes	54	83	422	86	1203	1187
	Regs	54	83	422	86	1203	1187
2^{nd} LSC	Nodes	3	2	1	2	3	3
	Regs	3	2	1	2	3	3
IN	Nodes	8	11	43	1	3	3
	Regs	8	11	43	1	3	3
OUT	Nodes	2257	5003	15943	12236	17670	20901
	Regs	119	63	92	62	108	249
INTENDRILS	Nodes	7	25	2	0	23	13
	Regs	0	0	0	0	0	0
OUTTENDRILS	Nodes	35	1	15	0	14	15
	Regs	35	1	15	0	14	15
TUBES	Nodes	1	1	0	0	0	0
	Regs	1	1	0	0	0	0
OTHERS	Nodes	19	0	2	0	3	2
	Regs	3	0	1	0	0	2

Table 4.3: Bow-tie decomposition of GRNs in different species. The regulators (denoted as Regs) are the nodes which have at least one outgoing edge in the extracted GRN. The 2^{nd} LSC refers to the next largest strong component separate from the LSC CORE.

sized GRNs, we have examined the variation of relative CORE size. This variation is clear in Figure 4.2A and especially in Figure 4.2B. The relative CORE size roughly increases as species complexity increases. This increase in percentage of network regulators in the bow-

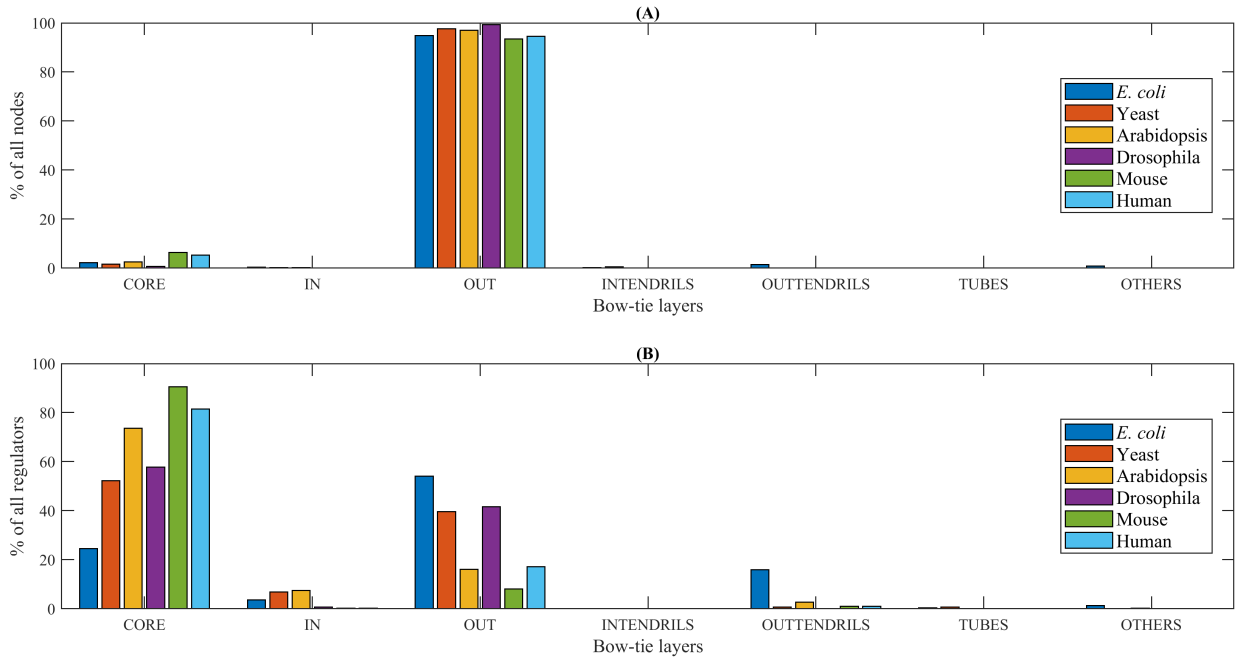


Figure 4.2: Bow-tie decomposition of GRNs. (A) Distribution of nodes in different bow-tie layers of GRNs in different species. (B) Distribution of regulators (nodes with at least one outgoing edge in the extracted GRN) in different bow-tie layers of GRNs in different species. A bow-tie architecture exists in all the GRNs. The CORE consists of a substantial percentage of all regulators. The relative CORE size generally increases with species complexity.

tie CORE in more complex organisms comes at the cost of a roughly decreasing percentage of regulators in the IN and the OUT layers, as can be observed in Figure 4.2B. Based on our observations, we can conclude that structurally the CORE size is a key differentiating factor in the bow-tie GRN architecture of different species, with a relatively larger CORE observed in more complex organisms.

To assess the effects of false positive and missing edges in the extracted GRNs on our observations, we perform sensitivity analysis experiments. In Figure 4.3 and Figure 4.4, we present the average distribution of nodes and regulators in the different layers from bow-tie decomposition of 1000 GRNs after random addition and deletion of 10% of the original

GRN edges respectively. On addition of edges, the size of the CORE increases. For Drosophila GRN with just 1 node in the IN layer, random edge addition leads to an incomplete bow-tie architecture, with the average number of IN nodes, rounded to an integer, being 0. Between species, the generally increasing trend in CORE size with complexity is still observed. The increase in the CORE size at the cost of the sizes of layers like the OUT would depend on factors like the network density and the original layer sizes, governing how a regulator node can now become part of the LSC, which can explain why we observe larger changes for some species in Figure 4.3. On deletion of edges, the CORE decreases in size, but is still substantially large and the roughly increasing trend in CORE size with complexity is preserved. There is an increase observed in the size of the OTHERS layer. The sensitivity analysis for much larger percentages (25% and 50%) of edge addition and deletion are also presented. Overall, these experiments suggest that, even with variations in the quality of the GRN data, analyzing these GRN architectures with the perspective of a bow-tie architecture with a LSC CORE makes sense, and there is a trend of increasing CORE size with species complexity.

Further, to quantify the extent to which the GRN bow-tie architectures are different than what would be expected simply by chance, we compared the bow-tie architectures observed in the empirical GRNs with their randomized counterparts. We looked at the LSC CORE size in these GRNs and the corresponding sizes in random networks having the same number and degree of nodes (Section 4.2.4). Figure 4.9 shows the LSC CORE layer sizes of 1000 random networks for every species, along with CORE size in the original GRNs. We observe that for *E. coli* and Yeast, the size of the CORE is smaller than that expected in similar random networks. As the species complexity increases in eukaryotes beyond Yeast, the size of the GRN bow-tie CORE is larger than expected in random networks. For Drosophila, most of the similar random networks do not have a full bow-tie architecture, with 0 nodes in the IN layer.

We present the sizes of the bow-tie LSC CORE as Z-scores with corresponding p-values

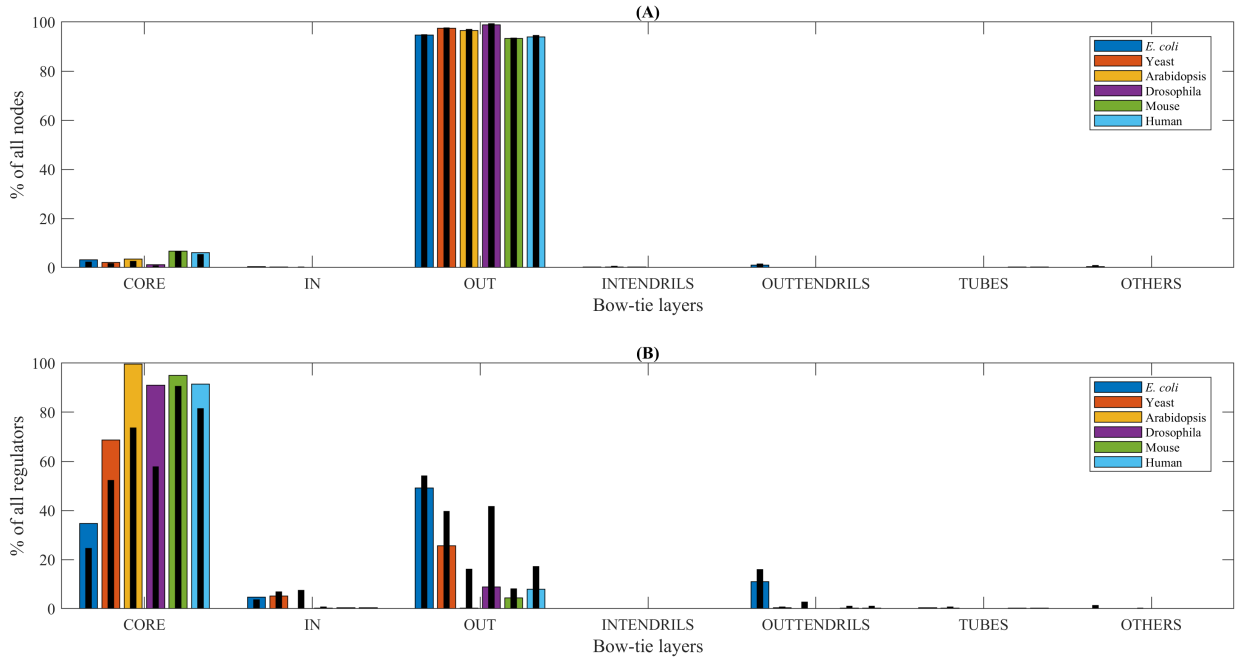


Figure 4.3: Bow-tie decomposition of GRNs after random addition of 10% edges. (A) Average distribution of nodes in different bow-tie layers. (B) Average distribution of regulators in different bow-tie layers. The original distribution of nodes and regulators are shown as black bars. The trend of increasing CORE size with species complexity is still observed.

in Table 4.4. For false discovery adjustment, we used the Benjamini-Hochberg procedure Benjamini and Hochberg, 1995. Using a false discovery rate of $Q = 0.15$, we can say that the GRN bow-tie LSC CORE size is significantly different from the LSC CORE size in similar random networks. Using a stricter false discovery rate of $Q = 0.05$, we find that the null hypothesis can be rejected for all species except *E. coli*. These comparisons show that the observed bow-tie architectures are characteristic features of these GRNs differentiating them from random networks of similar size and degree.

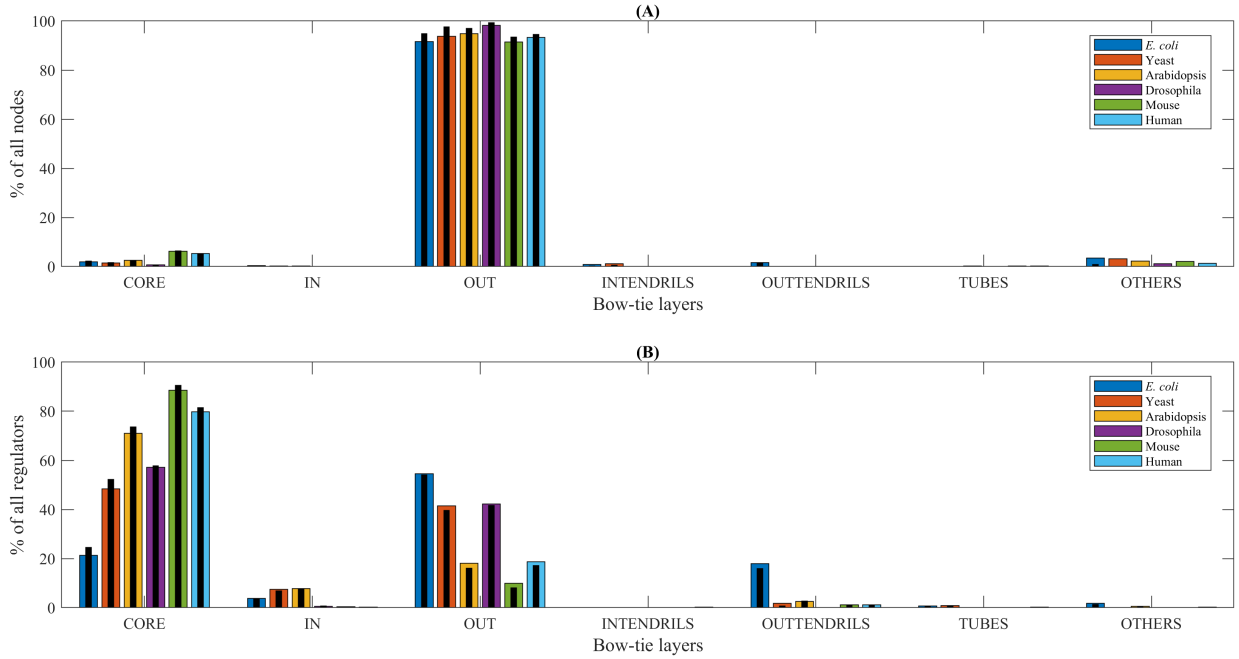


Figure 4.4: Bow-tie decomposition of GRNs after random deletion of 10% edges. (A) Average distribution of nodes in different bow-tie layers. (B) Average distribution of regulators in different bow-tie layers. The original distribution of nodes and regulators are shown as black bars. The CORE sizes are still substantial.

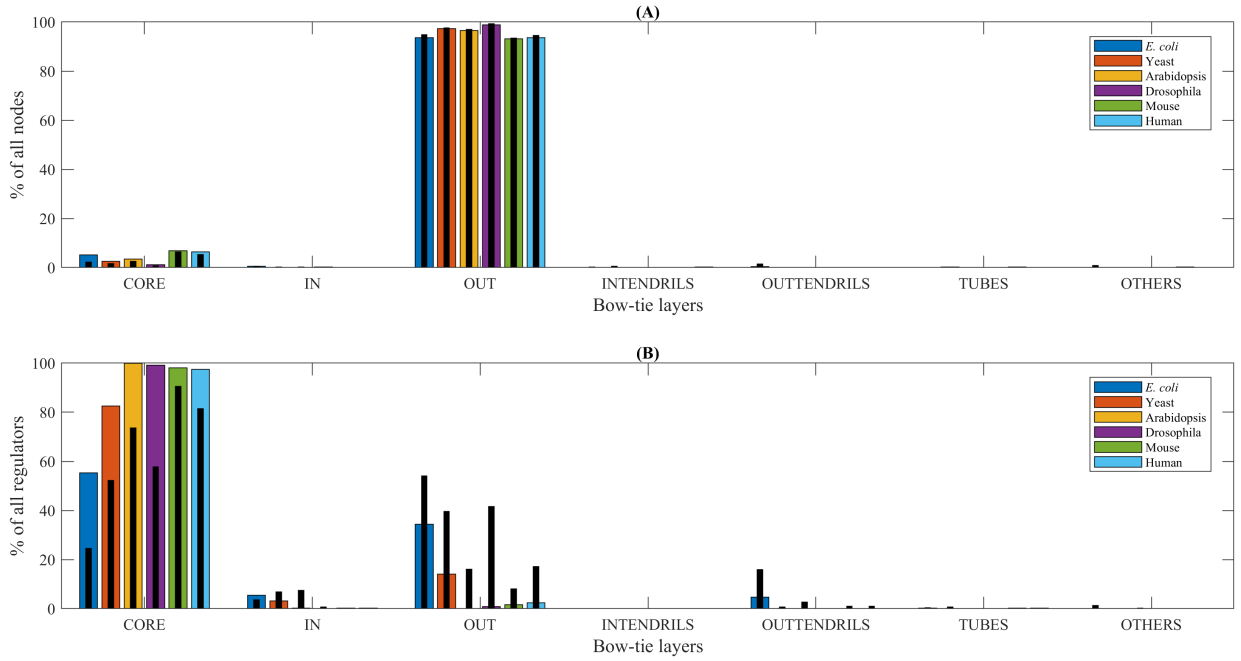


Figure 4.5: Bow-tie decomposition of GRNs after random addition of 25% edges. (A) Average distribution of nodes in different bow-tie layers (B) Average distribution of regulators in different bow-tie layers. The original distribution of nodes and regulators are shown as black bars. The trend of increasing CORE size with species complexity is not as clear as in the original distribution. The average number of IN layer nodes is 0 for Arabidopsis and Drosophila.

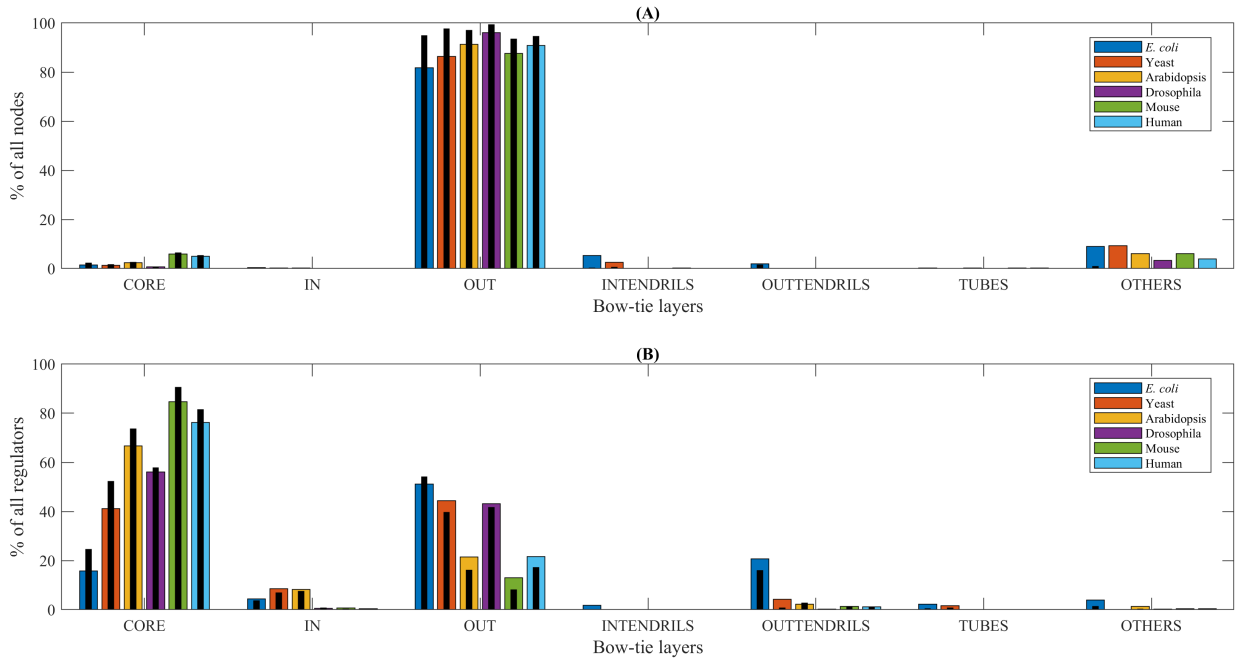


Figure 4.6: Bow-tie decomposition of GRNs after random deletion of 25% edges. (A) Average distribution of nodes in different bow-tie layers (B) Average distribution of regulators in different bow-tie layers. The original distribution of nodes and regulators are shown as black bars. The CORE size decreases for all species, but is still substantial. The OTHERS layer is larger.

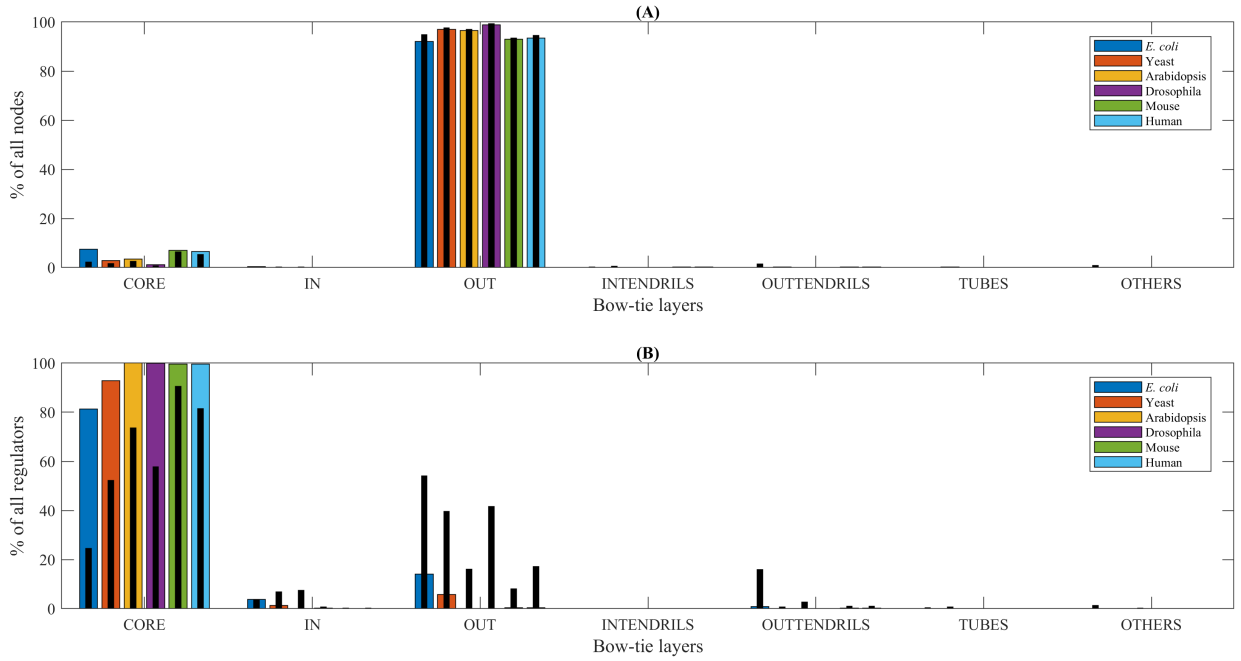


Figure 4.7: Bow-tie decomposition of GRNs after random addition of 50% edges. (A) Average distribution of nodes in different bow-tie layers (B) Average distribution of regulators in different bow-tie layers. The original distribution of nodes and regulators are shown as black bars. The trend of increasing CORE size with species complexity compared to the original distribution is more unclear. The average number of IN layer nodes is 0 for Arabidopsis, Drosophila and Human.

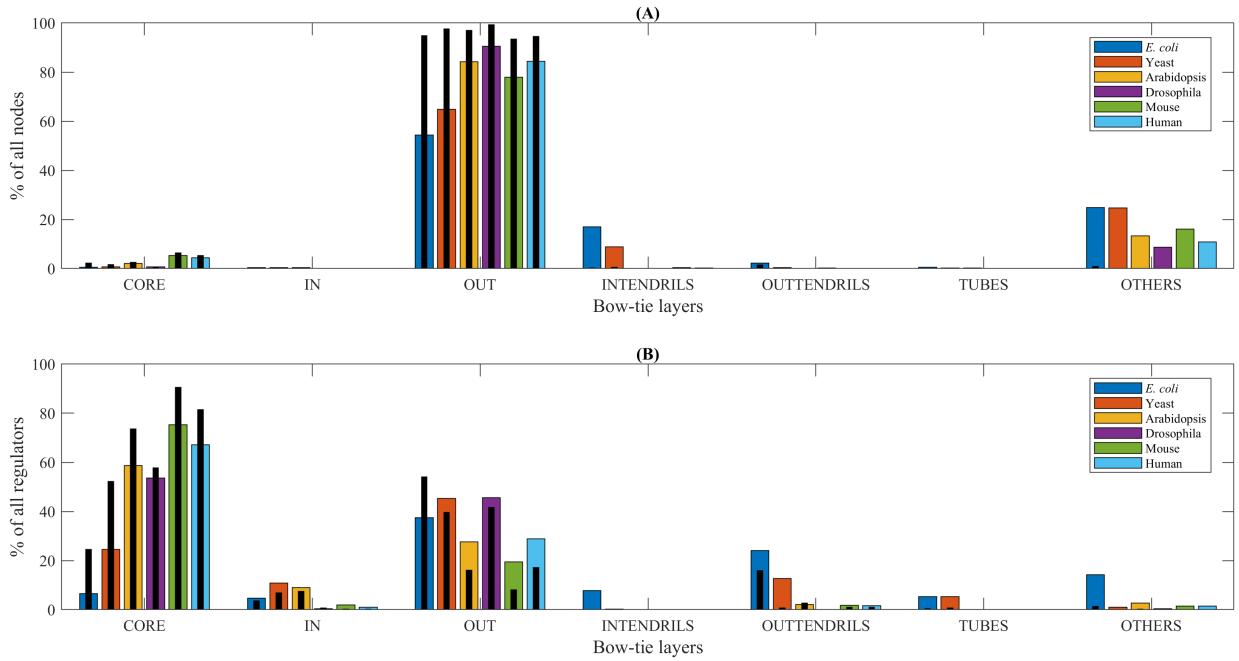


Figure 4.8: Bow-tie decomposition of GRNs after random deletion of 50% edges. (A) Average distribution of nodes in different bow-tie layers (B) Average distribution of regulators in different bow-tie layers. The original distribution of nodes and regulators are shown as black bars. The CORE is much smaller for all species but still mostly of substantial size. The OTHERS layer is much larger.

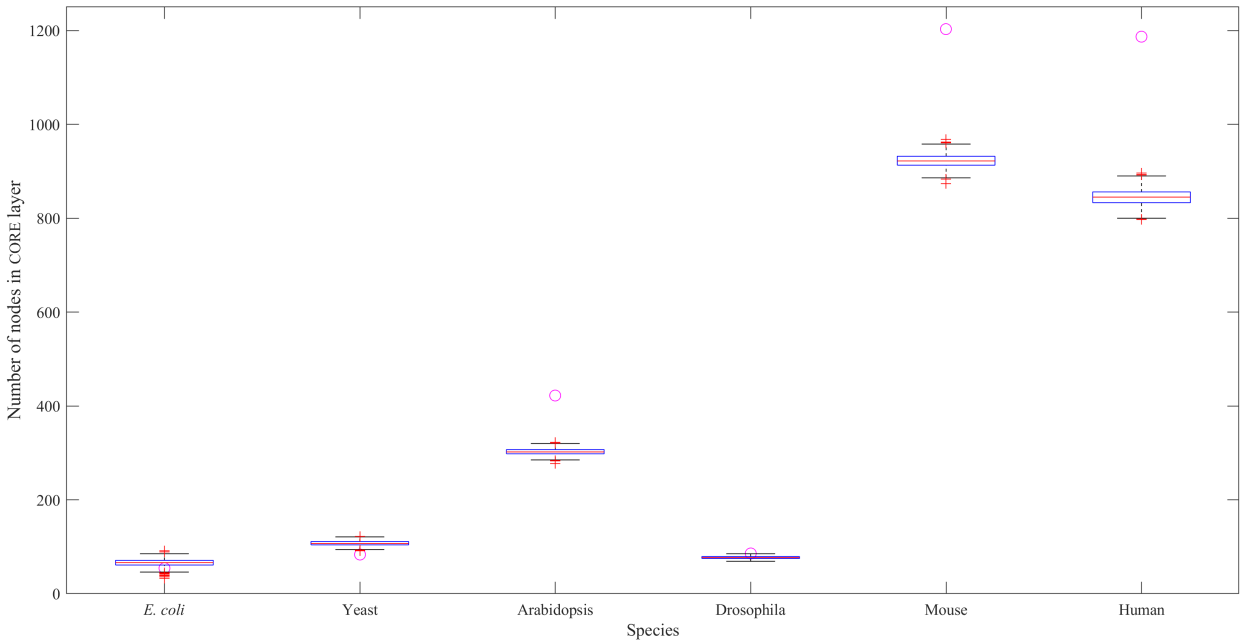


Figure 4.9: Bow-tie CORE sizes of similar random networks. Number of nodes in GRN CORE layer (circle) are compared to those in similar random networks (box plot) for different species. For *E. coli* and Yeast, the size of the CORE is smaller than expected in random networks. For species more complex than Yeast, the size of the CORE is larger than expected in random networks.

Species	Z-score	p-value	False discovery rate adjusted p-value
E. coli	-1.5560	0.1197	0.1197
Yeast	-4.7089	0.000002491	0.000003736
Arabidopsis	18.1863	0	0
Drosophila	3.6323	0.0002809	0.00033708
Mouse	20.6405	0	0
Human	19.7332	0	0

Table 4.4: Bow-tie CORE size comparison against similar random networks. Very small p-values are shown as 0. For false discovery rate adjustment we used the Benjamini-Hochberg procedure.

4.4 Discussion

4.4.1 Summary of observations

From our results in Table 4.3 and Figure 4.2, we find that a bow-tie architecture with a distinct LSC bow-tie CORE exists in the GRNs of all six species of varying complexity. There can be other perspectives of looking at these GRN architectures. Modularity and hierarchy are characteristics of GRNs (Hatleberg and Hinman, 2021), and a perspective would be to consider the hierarchy between these individual layers or modules. A small IN layer followed by a larger CORE and the OUT layer with the largest number of nodes resembles a pyramid shape. Prior work has shown the existence of a pyramidal hierarchical architecture in the GRNs of *E. coli* and Yeast (H. Yu and M. Gerstein, 2006). However, in contrast to H. Yu and M. Gerstein, 2006, in our study we find that the bottom or the OUT layer has TFs which regulate other TFs. Additionally, the strongly connected CORE between the IN and the OUT layers justifies the perspective of the bow-tie architectural feature in these GRNs. In the

bow-tie architectures, we observe a very small IN layer in some GRNs. Our analyzed general GRNs are not specific to any context, and for a species only a part of the general GRN is active for a single cell type. As such, and with a very small IN layer, it is possible that the GRN for particular cell types might not have the bow-tie architecture. In this study, we aim to look for the global architectural bow-tie feature in these general GRNs and how they differ between species, to be able to predict a trend in the emergence of a regulatory system property with varying biological complexity, and the predicted trend can be useful in those cell type-specific systems as well.

From our results, we observe that there is a general increase in bow-tie CORE size, relative to all nodes and all regulators in the GRN, with the complexity of the species. Our sensitivity analysis in Figure 4.3, Figure 4.4, Figure 4.5, Figure 4.6, Figure 4.7 and Figure 4.8 and comparison with similar random networks in Figure 4.9 and Table 4.4 show that the bow-tie architectures in these GRNs are characteristic features and can not be explained just by chance.

Our observations build on and add to the GRN architecture analysis results obtained from prior research. A bow-tie architecture with a distinct LSC CORE has been previously observed in the dynamical backbone of Yeast GRN (Rodriguez-Caso, Corominas-Murtra, and Solé, 2009) and in Arabidopsis TF-TF networks (S. Luo et al., 2018). However, the authors of (Rodriguez-Caso, Corominas-Murtra, and Solé, 2009) did not find a bow-tie architecture in the dynamical backbone of the analyzed *E. coli* GRN, with the LSC not much larger than the 2^{nd} LSC. The GRN consisted of 1607 nodes or about 36% of the species total genes. In contrast, with the use of a more complete GRN with greater than 50% of the total genes of the species, we observe a distinct LSC CORE between IN and OUT layers for the prokaryote *E. coli* and for other more complex eukaryotic species.

We observe an increase in bow-tie relative CORE size with the complexity of the

species, but this increase is not monotonic (Figure 4.2). A possible explanation for these slight variations from the trend of relative CORE size increase with complexity is variation in the GRN data quality from different data sources. Specifically in Figure 4.2A, a larger CORE size relative to all nodes is observed in *E. coli* than for more complex Yeast. There is also a subsequent drop for more complex species Drosophila. We believe that the likely cause of this is the incompleteness of the available GRN information in terms of the number of regulators in the extracted GRN. The percentage of regulator nodes out of all network nodes in the extracted GRN, where the corresponding absolute numbers are presented in Table 4.3, is highest for *E. coli* and lowest for *Drosophila*. This might contribute to the observed relatively high and low CORE sizes with respect to all nodes respectively for these two species. Therefore we validate the observation that the CORE becomes larger with complexity by also examining the size relative to all regulators in the GRN in Figure 4.2B. Here a clearer increase of CORE size with complexity is observed. The reason behind the slight drop observed here for Drosophila is probably that one of the two sources used by the curators of the Drosophila database (Section 4.2.1) has a stricter criterion of both binding and transcriptional regulation evidence for interactions. Our sensitivity analysis demonstrates that our results are quite robust to factors related to GRN data quality like incorrect or missing information.

4.4.2 Variation of controllability with complexity

Next, with our observations about the differences in GRN architectures between species, we aim to understand their biological implications. For that purpose, here we use previously proposed association of the dynamical system property of controllability with the bow-tie architecture and specifically its CORE layer size. This enables us to suggest hypotheses about how controllability may have emerged differently with biological complexity.

It has been proposed that a larger bow-tie CORE reduces controllability (Csete and Doyle, 2004). More complex organisms with more cell types should have more attractor states, as these attractor states of GRNs are considered to correspond to gene expression profiles associated with each cell type (S. Kauffman, 1969). We hypothesize that in such cases, perturbing the regulatory system to move from an undesired attractor to a desired attractor might be more difficult. This reduces the system controllability with complexity, that comes with a larger GRN bow-tie CORE, as observed for more complex species. Tighter control of the regulatory system may be related to more extreme conditions and less resources (Csermely et al., 2013), which might explain why less complex organisms including bacteria in our analysis have a smaller bow-tie CORE allowing more rigid control and support our hypothesis.

Additionally, we are able to suggest a complexity based division between species in terms of controllability. Comparison with random networks similar in size and degree distribution in Figure 4.9 reveals that the LSC CORE is smaller than expected by chance in *E. coli* and Yeast GRNs. Similar results for LSC size were previously observed in GRNs of *B. subtilis* and *E. coli* (Kumar et al., 2015), and Yeast (Jothi et al., 2009). For more complex eukaryotic GRNs, we observe that the bow-tie CORE size is larger than expected in similar random networks. So it is reasonable to speculate that for prokaryotic bacteria and unicellular eukaryotes living in comparatively more extreme conditions, greater regulatory system controllability is a key requirement. Our work has focused on how the GRN bow-tie architectures in these species have evolved to possibly support these requirements.

4.5 Strengths, limitations and directions

In this study we investigate the GRNs of several species and demonstrate the existence of a bow-tie architecture with a distinct LSC CORE in them. We show that the bow-tie is a characteristic GRN architectural feature. Among the strengths of our work, to our knowledge this is a novel comprehensive bow-tie architecture analysis of GRNs in several species of widely varying complexity. We further observe an increasing trend in relative CORE size with species complexity and hypothesize how the dynamical gene regulatory system property of controllability has emerged differently with complexity. The controllability of the gene regulatory system is very relevant in therapy, as cancer cells are considered to be trapped in abnormal attractor states (S. Huang, Ernberg, and S. Kauffman, 2009). Understanding how controllability emerges differently between species can lead to novel systems-based therapy approaches for diseases like cancer. Our work has provided valuable insights into the structural basis of this difference. For instance, the larger bow-tie CORE size for more complex organisms like Human needs to be taken into account in coming up with potential approaches for controlling the regulatory system state. Another possible benefit of our work is that the observed trends from the analysis of GRNs in several well-studied species can provide guiding directions for studies on less-studied or non-model species whose regulatory interaction information is largely incomplete at present.

A limitation of this work is that using other GRN data sources or a different set of GRN extraction criteria could affect our observations. For our analysis we depend on the information available in existing state of the art biological data sources, with GRN extraction criteria aimed at an optimal ground of comparison. Supported by our sensitivity analysis experiments, we believe our results are quite robust to data quality factors and hence the corresponding possible biological explanations hold merit. As new experimental methods for collecting data on regulatory interactions are developed, more complete and accurate data

on regulatory networks for more species should become available. We anticipate that the methods and results presented here will enable more detailed analysis of this data.

Future directions could aim at testing the hypotheses proposed in this work. It can be possible to quantify controllability in dynamical models, however, obtaining accurate dynamical models of these general GRNs of different species is a challenging problem on its own (Daniels et al., 2018; Cao and Grima, 2018). Metric definitions on real systems should be standardized. Our hypothesis of how controllability emerges differently with species complexity could then be verified, and the role of the bow-tie architecture CORE size difference can be assessed by possible *in vitro* GRN modification experiments. We need to consider other factors, including connectivity within and between different bow-tie layers, that might also govern controllability. However, for verifying the impact of the GRN bow-tie architecture in the proposed relationship, understanding how this architecture governs the network dynamics is of prime importance.

In our work we only look at the structural relationship of the GRN architecture with controllability, but in future we want to investigate the details of how the network architecture governs the network dynamics. For this we need to understand how an individual bow-tie layer governs the dynamics associated with that layer, and then possibly extend this to how the global bow-tie architecture controls the global network dynamics, within and between species. Determining how dynamical behavior associated with specific biological functions or pathways is controlled by the individual layers and the overall bow-tie architecture would provide new and valuable understanding of the functionality of GRNs. In our study we consider a general trend in one direction of the emergent property with increase of bow-tie CORE size in more complex species. However, detailed analysis of dynamics could reveal and explain the more complicated nature of this variation. The insights we provide here in our work can be useful for such future dynamical analysis.

4.6 Chapter Summary

In this chapter we present the work in our paper (Ghosh Roy, S. He, et al., 2021). We investigate the existence of an architectural feature – the bow-tie architecture in the GRNs of species of widely varying complexity, from prokaryotes to unicellular eukaryotes to multicellular eukaryotes. We obtain general GRNs not specific to any context from public databases and arrange them in an order of complexity defined on the basis of the number of cell types. From Table 4.3 and Figure 4.2, we find the existence of a bow-tie architecture with a distinct LSC CORE layer in all six analyzed GRNs. Sensitivity analysis in Figures 4.3, 4.4, 4.5, 4.6, 4.7 and 4.8 and comparison with random networks in Figure 4.9 and Table 4.4 demonstrates that the observed bow-tie architecture is a characteristic feature of GRNs and can not be explained just by chance. Additionally, we find a generally increasing trend of the relative CORE size with species complexity.

Based on these observations and using previously studied relationship of the bow-tie CORE size with the system property of controllability, we hypothesize how controllability has emerged differently with species complexity. We argue how a larger CORE size in more complex species comes with decreased controllability. We also discuss how prokaryotes and unicellular eukaryotes which survive in more extreme conditions require higher controllability, and these requirements are supported by their respective GRN bow-tie architectures. In this study we focus on the structural relationship between the bow-tie GRN architecture and the emergent property of controllability. However, to predict not just a trend but more specific values of this emergent property, we would need to understand how exactly the bow-tie architecture governs the network dynamics. For a large sized network, such modeling is an immensely complicated task with several parameters. When lower biological level measurements are to be mapped to emergent properties at higher levels, the number of associated parameters increases further making the modeling more complex.

Chapter Five

Visible Neural Network for Interpretable Prediction of Cancer-specific Survival Risk

In this chapter we aim to learn the mapping function that maps genotype to phenotype, that is lower-level network state measurements to higher organism-level emergent property of the survival risk, using the signaling network structure and how the known structure changes for a particular disease. Survival risk prediction using gene expression data is important in making treatment decisions in cancer. Standard neural network (NN) risk prediction models are black boxes with lack of interpretability. Interpretability denoting the degree to which a model's internal operations can be understood by a human, here in biological terms, makes the model more suitable for use in high-stakes clinical applications. In this chapter we will use the term architecture to refer to how the neurons in a neural network are connected with each other. More interpretable visible neural network (VNN) architectures are designed using biological pathway knowledge. But they do not model how pathway structure can change for particular cancer types. We propose a novel Mutated Pathway VNN or MPVNN architecture, designed using prior signaling pathway knowledge and gene

mutation data-based edge randomization simulating signal flow disruption. We assess the cancer-specific survival risk prediction performance of our MPVNN architecture compared to standard non-NN and similar sized NN survival analysis methods. We also interpret trained MPVNN architecture to obtain insights, and assess the reliability of some such insights using evidence in literature.

The organization of this chapter is as follows. Section 5.1 presents the background of our work. In Section 5.2 we present the risk prediction problem and our proposed MPVNN architecture and its interpretation methodology. Sections 5.3 and 5.4 present our experimental datasets and results respectively. We assess the reliability of MPVNN interpreted insights and discuss future directions in Section 5.5. The summary of this chapter is presented in Section 5.6.

5.1 Background and related work

Cancer is a leading cause of death worldwide, and a substantial amount of medical research is focused on survival analysis of cancer patients. The suitability of a particular treatment method could be guided by the predicted survival risk of the patient. The effectiveness of a treatment method could also be measured using the predicted risk. Gene expression data has been extensively used for cancer risk prediction, and different machine learning methods have been applied for this survival analysis task by learning from survival data (W.-Y. Cheng, T.-H. O. Yang, and Anastassiou, 2013). Neural networks (NNs) are an effective category of machine learning methods which have been used for gene expression-based cancer risk prediction (Katzman et al., 2018; Z. Huang et al., 2019).

One major challenge of some machine learning models including standard NNs is lack of interpretability. Model interpretability refers to the degree to which the model's internal

operations can be understood by a human (Biran and Cotton, 2017), which here denotes being understood in biological terms. This understanding would point to how particular biological entities and relationships are used internally by the model in mapping the input to the output. A more interpretable model would inherently increase a user's trust on the model, and hence be considered more suitable for use in high-stakes clinical applications (Rudin, 2019).

There are methods which can explain standard black box NNs. Examples of such methods are Layerwise Relevance Propagation (Bach et al., 2015), Integrated gradients (Sundararajan, Taly, and Q. Yan, 2016) and DeepLIFT (Shrikumar, Greenside, and Kundaje, 2017). DeepLIFT (Deep Learning Important FeaTures) method assigns importance scores to each input feature for a given output, by backpropagating the contribution of all neurons in the NN. As discussed in Rudin, 2019, for high-stakes applications, instead of trying to explain black box NNs, we should be coming up with NNs which are interpretable by design and hence after learning can be interpreted using straightforward approaches. Such interpretable NNs is the focus of our work.

Visible NNs or VNNs are neural networks where biological meanings are attached to intermediate neurons, and these VNNs provide increased model interpretability compared to standard black box NNs (M. K. Yu et al., 2018). Under a visible architecture, neurons represent biological entities like genes, proteins, pathways, cell subsystems, etc. and connections between the neurons represent biological relationships. Here we refer to those NNs as VNNs where neurons in each intermediate layer are associated with some explicit biological meaning.

An important source of biological knowledge for use in the design of VNNs is that of biological pathways. VNNs whose design uses pathway knowledge with neurons representing pathways include P-NET (Elmarakeby et al., 2021), GenNet (Hilten et al., 2020)

and DCell (J. Ma et al., 2018). These VNNs are designed from hierarchical knowledge and have multiple intermediate layers. Another VNN designed using pathway knowledge is knowledge-primed NN or KPNN (Fortelny and Bock, 2020), which is a NN architecture analogous to the structure of signaling pathways and also has multiple intermediate layers. A VNN with one intermediate layer, designed using protein–protein (PPI) and protein–DNA (PDI) interactions data, for predicting cell type from single cell expression values has been explored (Lin et al., 2017). However, none of these VNNs model how a known biological pathway structure can change for a particular disease.

5.2 Methods

5.2.1 Problem

We address the survival analysis task as a ranking problem of predicting the survival risk score. There are other approaches of addressing survival analysis. For example, in a classification setting, the patients can be stratified into risk categories (Y.-C. Chen, Ke, and Chiu, 2014). However, such a stratification based on a threshold like the median survival time might not be accurate in a patient population with a mix of cancer stages corresponding to widely varying medians. In another analysis approach, time-to-event distributions can be estimated (Chapfuwa et al., 2018). Here our aim is to predict risk scores indicative of the survival times so that ordering is correct. A clinician might be interested only in knowing if a particular treatment causes an increase in survival time without the need to know the exact value of the said time.

The survival data comprises two major components : the i^{th} patient survival time to event of interest t_i, and the censoring indicator l_i where a value of 1 denotes that the event

was observed and a value of 0 denotes censoring. In our risk prediction task, the objective is to predict a relative risk score $f(x_i)$ as output given the expression profile $x_i \in \mathbb{R}^N$ consisting of expression values of N genes as input, where a larger survival time can be associated with a larger risk score for cases that can be ordered.

Here we have used a standard survival analysis performance assessment metric of the Concordance Index (CI) or c-index (Harrell Jr, K. L. Lee, and Mark, 1996). It is a measure of agreement between the predicted survival risk and the observed survival. The CI is defined as follows

$$\text{CI} = \frac{1}{|\epsilon|} \sum_{(i,j)\in\epsilon} 1_{f(x_i)<f(x_j)}, \tag{5.1}$$

where $\epsilon = \{(i,j) \mid l_i = 1 \text{ and } t_j > t_i\}$. The CI metric is a value between 0 and 1, where a value of 0.5 denotes random prediction. The objective here is to maximize the CI value. Since the CI itself is not differentiable, we consider the following differentiable exponential lower bound on the CI (Steck et al., 2008):

$$\frac{1}{|\epsilon|} \sum_{(i,j)\in\epsilon} 1 - e^{f(x_i)-f(x_j)}. \tag{5.2}$$

We use the negative of this lower bound as the loss function for minimization. As in Wulczyn et al., 2020, during optimization we ignore the denominator of Eq. 5.2, and evaluate the loss over training batches.

The patient cancer survival outcome endpoint used in these experiments is also the disease-specific survival (DSS) endpoint (J. Liu et al., 2018). A DSS event denotes death specifically from the diagnosed cancer type, and the event time is measured from the date of initial diagnosis until that of event. The censored time denotes the time from the date of initial diagnosis until the date of death due to another cause or the date of last contact. This DSS endpoint is difficult to derive and is an approximation of the true DSS (J. Liu et al.,

2018). However, we want to be able to predict the cancer-specific survival risk, which has a more direct relationship with the cancer modeling. So we have selected DSS over a more commonly used endpoint – the overall survival (OS), and used the TCGA datasets for our experiments which provide this DSS data for multiple cancer types.

5.2.2 Proposed Architecture

To solve this problem of risk prediction from gene expression profiles, we propose a simple VNN architecture with one intermediate layer connecting the input genotype layer and the output phenotype layer. The input and the intermediate layers have N neurons each, and the output layer has one. The proposed MPVNN architecture is presented in Figure 5.1.

In the architecture, an intermediate layer neuron is assigned to represent the perturbation at one gene, where a gene is considered to be perturbed if it is in the path of actual flow of signal. Our usage of the term perturbation is in the context of signal flow, which is different from the notion of a gene being perturbed when mutated. Each intermediate layer neuron is connected to the input layer neuron representing the expression of the same gene. When using just pathway knowledge and no mutation data, each intermediate layer gene perturbation neuron is additionally connected to other input layer neurons representing the expression of genes which are its known neighbors in a signaling pathway. So a gene perturbation is derived from its own expression and the expression of its pathway neighbors, which are the genes with edges to or from the gene in consideration. This VNN design is unchanged by autoregulatory edges, or by cases where an edge is shared by two or more pathways or edges exist between two genes in both directions within or across pathways. This is the PVNN architecture, which is the version of the MPVNN architecture without the mutation data-based edge randomization.

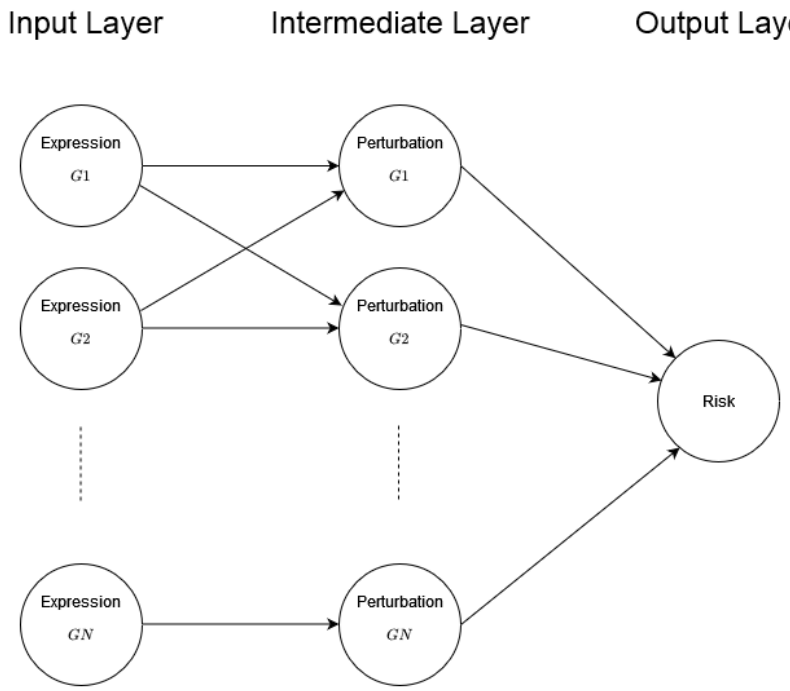


Figure 5.1: Proposed MPVNN architecture. Each input layer neuron represents the expression of a gene, and the output layer neuron represents cancer-specific survival risk. Each intermediate layer neuron is assigned to represent perturbation at a gene, where a gene is considered to be perturbed if it is in the path of actual flow of signal. A gene perturbation is derived from the expression of the same gene, and the expression of genes which are connected to the gene in consideration by either a known pathway edge or a randomly connected edge. For example, a known pathway edge or a randomly connected edge $G1 \rightarrow G2$ leads to two NN connections, one from input layer $G1$ expression neuron to intermediate layer $G2$ perturbation neuron, and a second from $G2$ expression neuron to $G1$ perturbation neuron. The randomly connected edges are obtained by replacing a fraction of known pathway edges using mutation data.

The MPVNN architecture is an extension of the PVNN architecture using another type of data, non-silent gene mutation data. The signal flow through the known pathway edges can be disrupted in cancer. Simulating the change in pathway structure for a particular cancer type, we use the additional gene mutation data for that cancer to replace a certain fraction of known pathway edges with random gene connections. These new connections are aimed at capturing signaling interactions not present in the used prior knowledge, which can be important in survival risk prediction for that particular cancer type. So the connections between the input layer and the intermediate layer neurons are used to represent the signaling edges, either obtained from prior knowledge or from the randomization using mutation data. The design algorithm in Algorithm 3 guides the connections between the input and the intermediate layers.

For a cancer type, we compute a mutation fraction per signaling pathway, which is a rough estimate of how much the signal flow in the pathway is disrupted. With the mutation fraction, we randomly replace this same fraction of the known pathway edges with new edges and connect the neurons accordingly. In this case, the intermediate layer neurons derive perturbations at a gene from:

- the expression of the same gene;
- the expression of some neighbor genes, as per known pathway knowledge;
- the expression of genes that it is randomly connected with, based on the mutation data.

For the MPVNN architecture design, the input is the list of all N genes and P pathways, the gene and the edge lists for every individual pathway and the mutation data for the N genes. We assume that over all the pathway edge lists, no autoregulatory edge exists, an edge and its reversed edge together do not exist, and an edge belongs to only one pathway.

Algorithm 3 MPVNN design algorithm.

Input: List of all N genes N_{all}; List of all P pathways P_{all}, Gene list N_p & Edge list E_p for every pathway $p \in P_{all}$; Mutation data $M \in \{0,1\}^{K \times N}$;
Output: Connection matrix $W \in \{0,1\}^{N \times N}$ denoting connections between input and intermediate layers
foreach gene $Gn \in N_{all}$ **do**
    Number of mutated samples $M_{Gn}^{sum} \leftarrow \sum_{k=1}^{K} M_{k,n}$
end foreach
Most mutated genes $M^{mut} \leftarrow \{Gm \ \forall \ M_{Gm}^{sum} \geq 0.99$ quantile of all non-zero $M_{Gn}^{sum}\}$
foreach pathway $p \in P_{all}$ **do**
    $frac_p \leftarrow \frac{|N_p|}{N}, \quad thr_p \leftarrow \frac{|N_p \cap M^{mut}|}{|M^{mut}|}$
end foreach
Connection matrix $W \leftarrow$ createW(N_{all}, N_p, E_p, thr_p, $frac_p$ for $p \in P_{all}$)

function CREATEW(N_{all}, N_p, E_p, thr_p, $frac_p$ for $p \in P_{all}$)
    $W \leftarrow I_N$
    $E_{new} \leftarrow \emptyset$
    foreach pathway $p \in P_{all}$ **do**
        foreach edge $a \rightarrow b \in E_p$ **do**
            $rnum_1 \leftarrow random()$
            if $rnum_1 < thr_p$ **then**
                do
                    $rnum_2 \leftarrow random()$
                    if $rnum_2 \geq thr_p$ **then**
                        $a, b \leftarrow$ Select 2 unique genes randomly from N_p
                    else if $rnum_2 < thr_p \frac{frac_{1 \neq p}}{1 - frac_p}$ **then**
                        if $random() \geq thr_{1 \neq p}$ **then**
                            $a, b \leftarrow$ 2 unique genes randomly from $N_{1 \neq p}$
                        else
                            $a, b \leftarrow$ 2 unique genes randomly from $N_{all} - N_{1 \neq p}$
                        end if
                        ⋮
                    else if $rnum_2 < thr_p \frac{frac_{1 \neq p} + \cdots + frac_{P \neq p}}{1 - frac_p}$ **then**
                        if $random() \geq thr_{P \neq p}$ **then**
                            $a, b \leftarrow$ 2 unique genes randomly from $N_{P \neq p}$
                        else
                            $a, b \leftarrow$ 2 unique genes randomly from $N_{all} - N_{P \neq p}$
                        end if
                    else
                        $a, b \leftarrow$ 2 unique genes randomly from $N_{all} - \sum_{p \in P_{all}} N_p$
                    end if
                while $(a \rightarrow b \in \cup_{p \in P_{all}} E_p) \vee (a \rightarrow b \in E_{new}) \vee (b \rightarrow a \in \cup_{p \in P_{all}} E_p) \vee (b \rightarrow a \in E_{new})$
                Add $a \rightarrow b$ in E_{new}
            end if
            $W_{ab} = 1$
            $W_{ba} = 1$
        end foreach
    end foreach
    return W
end function

This keeps the number of connections in PVNN and MPVNN same for fair performance comparison. The design algorithm output is the matrix $W \in \{0,1\}^{N\times N}$ denoting which connections exist between the N input layer gene expression neurons and the N intermediate layer gene perturbation neurons. From the mutation data, we compute the number of samples in which each gene Gn is mutated – M_{Gn}^{sum}. Then we obtain M^{mut} consisting of most mutated genes, each of whose above number is greater than or equal to the 0.99 quantile of all non-zero M_{Gn}^{sum}. For each pathway p with $frac_p$ fraction of total N genes, we compute a mutation fraction thr_p equal to the fraction of genes from this pathway in M^{mut}. Here we assume that the input gene expression profile does not consist only of genes all belonging to one single pathway, or we would use the PVNN architecture instead. For each known pathway edge, if a generated random number $rnum_1 \in [0,1)$ is $\geq thr_p$, we select the known edge, otherwise the edge is replaced and a new edge is selected.

For replacing a known edge with a new edge, another random number $rnum_2 \in [0,1)$ is generated. If this number is $\geq thr_p$, we select two pathway p genes randomly for a new edge. Otherwise, based on the value of the random number $rnum_2$, each remaining pathway $q \neq p$ is selected for consideration $thr_p \frac{frac_q}{1-frac_p}$ fraction of times. If the pathway q is selected for consideration, based on whether another new random number is $\geq thr_q$, two pathway q genes are selected randomly for a replacing edge, otherwise two genes are selected randomly from all genes not in pathway q. This is motivated by the design goal that pathways that have a high fraction of genes in most mutated genes and probably have greater disruption of signal flow, should contribute less to the genes of replacing edges. When no such pathway is selected dictated by the value of $rnum_2$, two genes, not belonging to any pathway in the list of pathways used in the design, are selected randomly for a replacing edge.

While obtaining a replacing edge in the MPVNN design, we only select two unique genes, and ensure that the selected replacing edge or its reversed edge do not belong either to the list of all known pathway edges or to the list of all already selected replacing edges.

This keeps the number of connections of MPVNN same with that of PVNN, and also can help in finding important signaling connections novel to what is present in the used prior knowledge. The connection matrix W is initially an identity matrix, and two entries where the row and column interchangeably denote the two genes of every selected edge, are made 1.

Though MPVNN architecture is generalized to use prior knowledge and mutation data for multiple pathways, here as a case study, we use one well-known cancer related pathway–phosphatidylinositol 3' -kinase (PI3K)-Akt signaling pathway in its design. It is a key regulator of processes involved in cell growth, proliferation, survival and apoptosis. These processes are tightly linked to the hallmarks of cancer (Hanahan and Weinberg, 2011). This pathway has been observed to play critical roles in various cancers (Jiang et al., 2020). As an example in this case study, we have assessed the effectiveness of our proposed architecture designed with the PI3K-Akt pathway obtained from the KEGG Pathway database (Kanehisa and Goto, 2000), in terms of predicting risk and interpreting which parts of the pathway are important in the prediction for different cancers.

5.2.3 Interpretation

Compared to standard black box NN survival analysis methods, a major benefit of our proposed VNN architecture is increased interpretability. Both the trained VNN architectures – PVNN and MPVNN can be interpreted to obtain top gene sets. These sets of genes within the larger signaling pathway are linked by flow of signal that is important in the prediction of survival risk for a particular cancer type, and are ranked in order of the importance associated with a set. The benefit of MPVNN over PVNN is that from interpretation we can obtain important signaling connections which are not present in the used prior knowledge.

Here we describe a straightforward mean weight amplitude-based VNN interpretation method, which fits with the signaling edge-based design of our VNNs. We first obtain the top gene perturbations in the intermediate layer, from the absolute weights connecting the intermediate layer gene perturbation neurons to the output neuron, averaged over the VNN runs. From each top gene perturbation, we find the top gene set connected by flow of signal associated with the gene perturbations.

The first candidate gene is the one with the top gene perturbation, and then the following process is repeated for every new gene added to the top gene set. For every candidate gene, we evaluate which of the gene expression neurons in the input layer, apart from the candidate gene itself and the last gene added to the top gene set in consideration, has the highest mean absolute weight connecting to the candidate gene perturbation neuron in the intermediate layer. We check whether the highest weight is ≥ 0.85 quantile of all the non-zero weights between the input layer and the intermediate layer neurons. If this new gene already belongs to the top gene set in consideration, we stop the process. Otherwise, we check whether the weight connecting the expression neuron for the newly selected gene to its perturbation neuron in the intermediate layer satisfies the weight threshold above. We additionally check if the perturbation neuron for the newly selected gene to the output layer phenotype neuron is ≥ 0.85 quantile of all the non-zero weights between the intermediate layer and the output layer neurons. Then this new gene is added to the top gene set. This whole process is done twice starting from the top gene perturbation to be able to possibly capture the important signal flow into and out of the gene.

The thresholds used are applied to ensure that genes with low connection weights and hence lower importance are not included in the top gene sets. We used a value of 0.85 quantile, which for the intermediate layer neurons, roughly translates to that out of the 1440 gene perturbations, around 200 can play some role of importance in risk prediction for a cancer type. The top gene sets obtained from the interpretation would depend on the values

of the thresholds used.

5.3 Experiments

Our experimental TCGA data for 10 cancer types is obtained from the UCSC Xena browser (Goldman et al., 2020). The data includes gene expression RNAseq data, binary gene-level non-silent mutation data and disease-specific survival data. We consider cancer samples, one per patient, which do not fall under the normal category in TCGA. For the expression profile, apart from the genes in the PI3K-Akt pathway, we also consider 1285 genes found to have systematic expression change in cancer (Torrente et al., 2016). Finally our patient expression profile consists of 1440 genes, for which both expression and mutation data are available.

The cancer types used in our experiments are presented in detail in Table 5.1. For each cancer type, the input-output data is split after a random shuffle into training (80%) and test (20%) sets with stratification on the censoring indicator/ event observed values. For the MPVNN architecture design, the available mutation data is used as a whole. For every machine learning method in our experiments, the input data features are standardized by mean subtraction and scaling to unit variance, based on those metrics computed from the training data.

We evaluated the performance of MPVNN against other comparable NN architectures. First there is the PVNN architecture which is designed using pathway knowledge and no mutation data-based randomization. In RaNN which is another randomized version of PVNN, the connections between the hidden layer and the input layer neurons are randomly shuffled. This represents a same sized NN with the same number of connections but designed without using prior knowledge or additional mutation data. We evaluated the performance

Cancer	Description	Number of samples	Events observed	Censored
BLCA	Bladder urothelial carcinoma	390	119	271
BRCA	Breast invasive carcinoma	1070	83	987
COADREAD	Colon and rectum adenocarcinoma	354	41	313
GBM	Glioblastoma multiforme	147	113	34
HNSC	Head and neck squamous cell carcinoma	493	130	363
KIRC	Kidney renal clear cell carcinoma	522	109	413
LIHC	Liver hepatocellular carcinoma	361	79	282
LUNG	Lung squamous cell carcinoma and adenocarcinoma	914	199	715
OV	Ovarian serous cystadenocarcinoma	272	150	122
STAD	Stomach adenocarcinoma	386	95	291

Table 5.1: Our experimental datasets for cancer-specific survival risk prediction.

of the fully connected artificial neural network or ANN with the same number of neurons. We have also compared the performance of these NN methods with a standard survival analysis method – the semi-parametric Cox Proportional Hazards (Cox-PH) model (Cox, 1972). We drop the expression values of those genes which give an initial warning of having very low variance in the Cox PH regression fitter.

We used a 4-fold cross-validation on the training set to identify the optimal hyperparameters, which is different from the approach followed in our classification experiments. The optimal hyperparameter setting for a particular method for a particular cancer type

is selected from a certain number of random searches on the list of given hyperparameter values. For this number, we used around 10% of the total number of possible combinations. Hence, the number of random searches was set to be 300 for NNs and 10 for Cox-PH model. The list of values to choose from for NNs is given as follows :

1. Activation function : tanh, sigmoid, ReLU
2. Optimizer : Adam, SGD
3. Learning Rate : 0.1, 0.01, 0.001
4. Batch Size : 16, 32, 64, 128
5. Epoch : 10, 25, 50, 100
6. Regularizer : L1, L2
7. Regularizer parameter : 0.0, 1e-5, 1e-3, 1e-2, 1e-1, 1.0.

For every NN, we used the same list of allowed epoch values. However, a larger number of epochs could improve performance for ANN with larger number of connections, and this is a limitation in our experiments. The hyperparameters for the Cox-PH model to select from are penalizer values in the set

$$\{0.0, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0\}$$

and the L1 ratio values between 0 and 1 in intervals of 0.1. For randomized architectures MPVNN and RaNN, the optimal connection matrix is also selected from the same 300 random searches.

The performance metric used is the CI value on the hold-out test set. We present the mean and the standard deviation from 20 runs. The data and code are available at https://github.com/gourabghoshroy/MPVNN.

Cancer	Cox-PH	ANN	RaNN	PVNN	MPVNN
BLCA	0.6919±0.0000	0.7455±0.0107	**0.7463±0.0046**	0.7130±0.0089	0.7214±0.0029
BRCA	0.4810±0.0142	**0.6507±0.0174**	0.5434±0.0082	0.6367±0.0151	0.6298±0.0040
COADREAD	0.5868±0.0000	0.7731±0.0673	0.7658±0.0474	0.6085±0.0230	**0.7804±0.0267**
GBM	**0.5822±0.0041**	0.5153±0.0521	0.5762±0.0421	0.4991±0.0168	0.5763±0.0115
HNSC	0.4912±0.0006	0.5864±0.0092	0.5920±0.0091	**0.6498±0.0025**	0.5827±0.0081
KIRC	0.7747±0.1159	0.8374±0.0257	**0.8474±0.0063**	0.7955±0.0049	0.8082±0.0063
LIHC	0.5549±0.0042	0.7527±0.0292	0.7600±0.0060	**0.7918±0.0052**	0.7878±0.0050
LUNG	**0.6644±0.0000**	0.5837±0.0124	0.5649±0.0035	0.6248±0.0031	0.6160±0.0057
OV	0.5895±0.0305	0.5886±0.0334	0.5869±0.0130	0.6102±0.0068	**0.6248±0.0050**
STAD	**0.7429±0.0000**	0.6207±0.0516	0.6699±0.0087	0.5502±0.0395	0.6800±0.0062
MACRO-AVERAGE	0.6160±0.0170	0.6654±0.0309	0.6653±0.0149	0.6480±0.0126	**0.6807±0.0081**
MACRO-AVERAGE WEIGHTED RANK	0.0921	0.0427	0.0428	0.0601	**0.0273**

Table 5.2: Cancer-specific survival risk prediction performance evaluation of MPVNN. The mean and standard deviation of the CI metric from 20 runs are shown for each cancer type, and then these values are macro-averaged over all cancer types. The macro-average weighted ranks are also presented, where the weighted rank is calculated as the difference between the maximum of mean CI metric values for all methods and the mean CI metric value for a particular method in a cancer type. The mean best, that is the method with the highest mean value, individually and macro-averaged, and the one with the lowest macro-average weighted rank, are all marked in boldface.

5.4 Results

In Table 5.2 we present the results of our MPVNN architecture for cancer-specific survival risk prediction for each cancer type, and finally these values macro-averaged over all cancers. We also show each method's macro-average weighted rank, where the weighted rank is calculated as the difference between the maximum of mean CI metric values for all methods and the mean CI metric value for a particular method in a cancer type. The lower the macro-average weighted rank, the closer to the best mean performance a method's mean performance is on an average across cancer types.

Our results show that MPVNN has better overall mean cancer-specific risk prediction

performance compared to the other methods that were tested. Firstly, its mean performance is better than that of the standard Cox-PH survival analysis model macro-averaged across all cancer types. Second, its overall mean performance is better than those of other comparable NN architectures – PVNN, RaNN and ANN. Compared to the next best method, MPVNN on macro-average has a mean CI metric higher by 0.015. Even though MPVNN is not the best in all of the 10 cancer types, its macro-averaged metrics suggest that the incorporation of signaling pathway knowledge and gene mutation data-based edge randomization in the VNN design can improve the overall mean prediction performance.

We compare the results of MPVNN with the overall second best ANN using Wilcoxon rank sum test (Mann and Whitney, 1947) p-values given in Table 5.3. For false discovery adjustment, we used the Benjamini-Hochberg procedure Benjamini and Hochberg, 1995. Using a false discovery rate of $Q = 0.05$, we can say that the performance of MPVNN is significantly different from that of ANN for 8 of the 10 cancer types. This might not be evident from the macro-averaged metric values as ANN is better than MPVNN in some cancers and worse in some. The significant difference with the higher overall mean performance points to the benefit of using MPVNN in terms of risk prediction.

Cancer	p-value	False discovery rate adjusted p-value
BLCA	6.3761e-08	6.3761e-07
BRCA	0.0003	0.0004
COADREAD	1	1
GBM	0.0002	0.0003
HNSC	0.2443	0.2714
KIRC	0.0001	0.0002
LIHC	0.0002	0.0003
LUNG	8.4416e-08	4.2208e-07
OV	1.5735e-05	5.2451e-05
STAD	0.0001	0.0003

Table 5.3: Performance comparison of MPVNN with fully connected ANN. The p-values are obtained from Wilcoxon rank sum test. For false discovery rate adjustment we used the Benjamini-Hochberg procedure.

To demonstrate the interpretability of our proposed MPVNN architecture, we have shown two top gene sets obtained from MPVNN interpretation in Figure 5.2. The top gene sets are given for 2 cancer types, ovarian and liver, in which the VNNs outperform other methods (MPVNN performs the best and the second best). We have selected the top gene set which has the first occurrence of any signaling connection that is not present in the used PI3K-Akt pathway edge list. For ovarian cancer, the top gene set consists of these genes : GNB3–PPP2R1B–FGF2→**NGFR**–PPP2R2B→AKT1→CHUK, where the top gene perturbation is marked in bold. The pathway edges are marked by arrows and the novel signaling interactions are marked by dashes. The shown top gene set for liver cancer comprises ANGPT2→FLT3–**FLT4**←ANGPT2. An important novel connection between genes FLT4 and FLT3 in the same gene group RTK is interpreted from the MPVNN architecture,

however these 2 genes are not connected by any given PI3K-Akt signaling path.

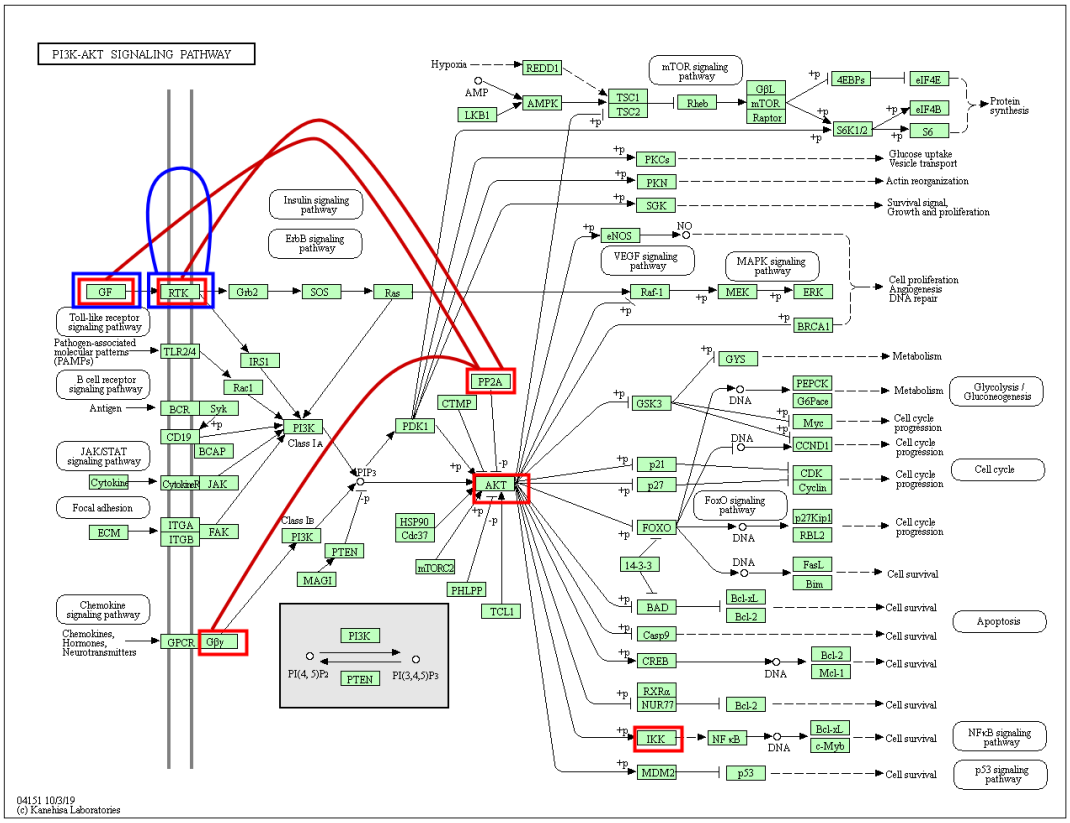


Figure 5.2: Top gene set from MPVNN interpretation for Ovarian (OV) and Liver (LIHC) cancers shown in red and blue respectively. The top gene set with the first occurrence of any signaling connection that is not present in the PI3K-Akt pathway edge list is selected for each cancer. With the PI3K-Akt pathway edges as arrows and the novel connections as dashes, the genes (and their groups) are as follows — Ovarian : GNB3(G$\beta\gamma$)–PPP2R1B(PP2A)–FGF2(GF)→NGFR(RTK)–PPP2R2B(PP2A)→AKT1(AKT)→CHUK(IKK), Liver : ANGPT2(GF)→FLT3(RTK)–FLT4(RTK)←ANGPT2(GF). In the PI3K-Akt pathway diagram (Kanehisa and Goto, 2000), the gene groups are bordered by colored rectangles and the novel connections are shown by colored curved lines for each top gene set.

5.5 Discussion

From our results, we find that our proposed MPVNN architecture has better overall cancer-specific survival risk prediction mean performance than other survival analysis methods. However, the mean performance of MPVNN is not the best in every single cancer type, given the corresponding data. The Cox-PH model is found to be a better risk prediction model than the NNs in some cancer types, though MPVNN is observed to be better on macro-averaged metrics. The high importance of the PI3K-Akt pathway in risk prediction is highlighted for those particular cancer types where PVNN performs the best. We also observe that for some cancers, the randomized RaNN or the fully connected ANN gives the best mean CI metrics. Since these two architectures lack interpretability, it is more difficult to understand in biological terms how they work. When RaNN is the best, one possibility is that there are other pathways involving the input genes, which are more effective in risk prediction. Incorporating knowledge and mutation data for multiple pathways in the architecture design might be helpful for performance. As described previously, our VNN architectures can be applied to use multiple pathways. For the cancer where ANN is the best, one possibility is that pathways which can play significant roles in risk prediction are not well represented in the input dataset genes, and a larger set of input genes, along with multiple pathway knowledge and mutation data in the architecture design, might improve the performance.

A major benefit of our MPVNN architecture is increased interpretability compared to black box NN methods, making it more reliable for use in clinical survival analysis. To validate some of the insights we obtained from MPVNN interpretation, we looked for literature evidence. In the top gene set for ovarian cancer, we observed important signaling interactions within the PI3K-Akt pathway between parts of the growth factor, the chemokine, the Akt and the NFκB signaling pathways. Some of these interactions are novel to what

exists in the used prior knowledge. Signal flow involving these four pathways together being important in the context of higher mortality in ovarian cancer has been previously indicated (Dong et al., 2013; Son et al., 2013).

In the top gene set in liver cancer, we observed important pathway edges ANGPT2→FLT3 and ANGPT2→FLT4 and a novel interaction between RTK genes FLT3 and FLT4. Connection to ANGPT2 from a RTK gene via calcineurin-NFAT has been studied in lung metastasis (Minami et al., 2013). So such a signaling path from any of the RTK genes FLT3 or FLT4 back to ANGPT2 can correspond to the interpreted novel interaction. Among the targets of the popular drug Sorafenib, which has been used effectively in the treatment of advanced liver cancer for over a decade, are both the FLT3 and FLT4 genes (Marisi et al., 2018). Interestingly, in an extensive Sorafenib trial (Llovet et al., 2012), ANGPT2 was found to be an independent survival predictor in the overall and the drug treated population, but not related with drug induced survival benefit. Compared to the baseline concentration, ANGPT2 concentration after 12 weeks of treatment was observed to increase in the placebo group, however it remained almost unchanged for treatment with the drug. So it is possible that the signal flow in the MPVNN interpreted top gene set is disrupted during treatment with Sorafenib by targeting both FLT3 and FLT4 genes, as a result of which the ANGPT2 concentration remains almost constant. Later the signal flow involving these genes probably again comes into effect.

Based on the above literature validation, we argue that MPVNN interpreted insights are reliable, pointing to signal flow that have critical roles in controlling cancer-specific survival risk. Further validation of these insights would require experimental verification, which is outside the scope of this study. These insights are flexible enough to include important signaling connections that are not present in the used prior knowledge. MPVNN interpreted insights can have correspondence with the mode of action of existing drugs like for Sorafenib in liver cancer above, and with further experimental studies in future can provide directions

for novel and more effective single or combinatorial drug therapy.

Edge randomization based on additional mutation data, which models how a pathway structure can change for a particular cancer type and replaces known edges with new connections, is a novelty of our MPVNN architecture. Heteroscedastic dropout with privileged or additional information is used in Lambert, Sener, and Savarese, 2018, although this differs substantially from MPVNN. Dropout for a NN architecture is used only during learning and not in prediction, whereas the edges in the MPVNN architecture are randomized at the beginning, before learning and subsequent prediction. Also in MPVNN, the use of the additional mutation data in the supervised prediction setting is unsupervised, without the requirement of being related to the input-output training data. This makes the architecture more robust to bias that may arise in supervised machine learning when some population groups are not well represented in the training set, for instance due to lack of survival data here, but mutation data is available for those groups and can be incorporated in the model.

As only the PI3K-Akt pathway used in our work is not sufficient in fully demonstrating MPVNN predictive power, future directions would focus on conducting larger scale experiments with larger gene expression profiles using knowledge and data for multiple pathways in the MPVNN architecture design to further evaluate cancer-specific survival risk prediction performance. In future, we would also like to experimentally investigate the roles of the important signal flow identified from MPVNN interpretation for different cancer types, and importantly explore their relevance to cancer drug target identification.

5.6 Chapter Summary

In this chapter we present the work in our paper (Roy et al., 2022), where we propose a novel Mutated Pathway Visible Neural Network or MPVNN architecture for predicting emergent

cancer-specific survival risk from patient gene expression data. For this prediction task, this MPVNN architecture uses knowledge of signaling network structure and models how the structure changes for a particular cancer type. In this visible neural network architecture, each neuron in the single intermediate layer between the input gene expression layer and the output risk phenotype layer is assigned to represent a gene perturbation. The connection between the input layer and the intermediate layer neurons represent known edges of a signaling pathway or random gene connections based on additional gene mutation data.

In our case study we use the PI3K-Akt signaling pathway with important roles in cancer to design the visible architecture. Our experimental results addressing survival risk prediction as a ranking task in Table 5.2 suggest that the MPVNN architecture can perform better than other similar sized NN and non-NN survival analysis methods. Our visible MPVNN architecture is more interpretable than standard black box NNs, and hence is more reliable for use in clinical survival analysis. Interpretation of the trained MPVNN architecture can provide insights about gene sets linked by flow of signal which are important in cancer-specific risk prediction, as shown in Figure 5.2. We assess the reliability of such insights for some cancer types using literature evidence, and argue that these insights correspond to the actual emergence of risk and can provide directions for drug target identification with further studies.

Chapter Six

Conclusions

This chapter summarizes what we learn from the overall work done in the thesis, and discusses some examples of future work. We have discussed some strengths, limitations and future directions of individual studies in respective chapters. Here we put forward these aspects of our work in the context of the central thesis objective.

6.1 Summary

The central objective of this thesis is to predict emergence in biological systems using architecture of the molecular networks used to model such systems. We make advances towards this objective through our research contributions addressing each of the three research questions.

In Chapter 3 we learn how to better infer GRN architecture with its aspects that can be useful in the prediction of emergence. Many popular GRN inference methods do not infer edge signs. Our proposed algorithm PoLoBag can infer the signed architecture of GRNs from a general form of gene expression data without any prior time course or gene knock-out assumptions or availability of reference wild-type measurements. Signed inference methods

based on only Lasso method suffer from some limitations, and we address them in our novel algorithm that combines Lasso models in a bagging setting and also uses polynomial features. PoLoBag algorithm gives more accurate signed inference than state-of-the-art methods Banjo and SIREN. Also unlike these other algorithms, PoLoBag infers edge directions and cycles, which along with edge signs are GRN architectural aspects useful in predicting emergence. A limitation of our algorithm is that we do not consider any time relationships in the expression data. However, dynamical time series data can help in better inferring causality in the form of GRN edge directions.

In Chapter 3 we next see the usefulness of signed GRN architecture in predicting emergent states of the gene regulatory system. Combining signed GRN architecture with dynamical information in our proposed dynamical K-core method, we find a trend towards our method better identifying top regulators in GRNs which can better predict emergent states of the regulatory system, compared to the standard K-core and random selection methods. We observe that the best mean prediction performance metrics are obtained by maximum out-degree regulator ranking. This points to that in predicting emergence, it is very important to consider how a node fits in the network, with regards to its local connectivity to other nodes and to the global architectural organization in the network.

In Chapter 4 we learn that the GRNs of prokaryotic bacteria to unicellular Yeast to multicellular human have a bow-tie architecture with a distinct largest strongly connected CORE layer. A bow-tie architecture has been previously observed in some eukaryotes. However, an investigation of this feature in GRNs of species of such widely varying complexity has not been performed prior to our work. We find that the observed bow-tie architectural feature is a characteristic feature of GRNs and can not be explained just by chance. Additionally, we observe a generally increasing trend in the CORE size with species complexity, and using previously studied relationship of the bow-tie CORE size with the system property of controllability, we predict a trend in the emergence of controllability with varying species

complexity.

We hypothesize how a larger CORE size observed in the GRNs of more complex species comes at the cost of reduced controllability. We argue that for less complex bacteria and Yeast which survive in more extreme conditions with less resources, higher controllability is needed, and these respective requirements are supported by the respective GRN bow-tie architectures. A limitation of this work is that we consider the structural relationship between the bow-tie architecture and the dynamical emergent property of controllability. To predict not just general trends but more specific values of controllability, we would need to model how the bow-tie architecture governs the dynamics, which for these large networks is a very complex task with many parameters.

In Chapter 5 we see how our proposed visible neural network MPVNN, which uses knowledge of signaling network architecture and additional mutation data-based edge randomization that models how known signaling network architecture can change for particular cancer types, predicts organism-level emergent phenotype of the cancer-specific survival risk from gene expression profiles of patients. We address the problem as a ranking task of predicting a risk score indicative of the survival time and suggest that overall MPVNN has improved mean prediction performance compared to similar sized NN and standard non-NN survival analysis methods.

Our proposed MPVNN has increased interpretability compared to standard black box NNs, and hence is more reliable for use in clinical risk prediction. We find out how interpretation of the trained MPVNN points to sets of genes connected by flow of signal, that are important in cancer-specific survival risk prediction. Using evidence from literature we argue that these insights are reliable, corresponding to the actual emergence of the risk phenotype. A limitation of our work is that only the PI3K-Akt signaling pathway was used in our experiments as a case study and this is not sufficient in fully demonstrating the

predictive power of MPVNN.

To summarize what we achieve through our work in this thesis, we first fulfill the objective of better inferring the GRN architecture with its aspects like the edge signs that are found to be useful in predicting emergence in the gene regulatory system. Next we are able to predict a trend in the emergent property of controllability of the gene regulatory system based on observed quantitative differences in the bow-tie architectural feature in the GRNs of species of widely varying biological complexity. Finally we achieve our goal of effectively predicting an organism-level emergent phenotype of cancer-specific survival risk from gene expression data in an interpretable way using signaling network architecture and modeling how the known architecture changes for particular cancer types. Overall, this is how we have worked towards the central objective of our thesis.

6.2 Future Work

In our PoLoBag algorithm we are able to infer signed GRNs from a general form of gene expression data without any time course assumptions. If some time series measurements are available, we do not assume any time relationships in the data, considering every time point as a separate steady-state measurement condition. Though our PoLoBag can provide edge directions along with edge signs, dynamical time series data can help in better inferring causality. It is possible to adapt ensemble regression-based methods like GENIE3 to infer unsigned networks from steady-state and dynamical data, considering the time dependence in the latter (Huynh-Thu and Geurts, 2018). So a very promising area of future work is coming up with an extension of our ensemble regression-based PoLoBag algorithm that can infer signed GRN from both steady-state and dynamical data. This will preserve the ability of our algorithm to infer GRN architecture from a general form of expression data, in the

absence of dynamical data, and impart the ability to better infer edge directions that can be useful in improved prediction of emergence, in the presence of dynamical data.

In our study we focus on the structural relationship between the bow-tie architecture and the dynamical emergent property of controllability to predict a general trend in the emergence of the latter. Future work based on our study should look at modeling how exactly the bow-tie GRN architecture controls the network dynamics to predict more specific values of this property. Dynamical modeling of large-scale GRNs is a very complicated task. We can possibly look at simplified approaches where we analyze how an individual bow-tie layer governs the dynamics associated with that layer, and then possibly extend this to how the global bow-tie architecture controls the global network dynamics. A suitable choice would be the CORE layer of the bow-tie. We could start with less complex organisms like *E. coli* and Yeast, though the dynamical modeling for 50 or 80 node layers would still be a very challenging task. An approach we can use is hierarchical decomposition of the bow-tie CORE into further internal layers using K-core decomposition method like in our work in Section 3.5, and look at hierarchically modeling the dynamics associated with the bow-tie CORE by starting with the dynamics associated with the innermost layer.

In future we could look at cancer-specific survival risk prediction by conducting larger scale experiments with larger gene expression profiles using architecture knowledge and mutation data for multiple pathways in the MPVNN neural network design. Future work can also focus on an extension of MPVNN for prediction of another important emergent phenotype of drug response in cancer cells. Effectively predicting drug response can address the issue of high failure rate of new drug candidates in clinical trials. An example of a drug response prediction model is DrugCell, which is a combination of a VNN for the input genotype data and an ANN for the input drug chemical structure data in two separate branches (Kuenzi et al., 2020). We can possibly look at how an interpretable VNN as an extension of our MPVNN maps both genotype and drug structure data together in one branch and

predicts response of the drug in the output. The data required for training and testing such a model is available in public data sources. An interpretable prediction model improves reliability and can lead to further insights on the mechanisms of drug response. A challenge in making such a neural network more interpretable is the need for prior knowledge regarding genes that can potentially be affected based on individual elements of chemical structure representation or fingerprint.

Appendix One

Useful Resources

Source Code and Data in Thesis

- PoLoBag in Chapter 3: https://github.com/gourabghoshroy/PoLoBag
- Bow-tie GRN architecture in Chapter 4:
  https://github.com/gourabghoshroy/Bow-tieGRN
- MPVNN in Chapter 5: https://github.com/gourabghoshroy/MPVNN

References

Adami, Christoph, Charles Ofria, and Travis C Collier (2000). "Evolution of biological complexity". In: *Proceedings of the National Academy of Sciences* 97.9, pp. 4463–4468.

Aibar, Sara et al. (2017). "SCENIC: single-cell regulatory network inference and clustering". In: *Nature methods* 14.11, pp. 1083–1086.

Akhshabi, Saamer and Constantine Dovrolis (2011). "The evolution of layered protocol stacks leads to an hourglass-shaped architecture". In: *Proceedings of the ACM SIGCOMM 2011 Conference*, pp. 206–217.

Alon, Uri (2007). "Network motifs: theory and experimental approaches". In: *Nature Reviews Genetics* 8.6, pp. 450–461.

Altschul, Stephen F et al. (1997). "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs". In: *Nucleic acids research* 25.17, pp. 3389–3402.

Anand, Amitesh et al. (2020). "OxyR is a convergent target for mutations acquired during adaptation to oxidative stress-prone metabolic states". In: *Molecular biology and evolution* 37.3, pp. 660–667.

Arendt, Detlev et al. (2016). "The origin and evolution of cell types". In: *Nature Reviews Genetics* 17.12, pp. 744–757.

Azeloglu, Evren U and Ravi Iyengar (2015). "Signaling networks: information flow, computation, and decision making". In: *Cold Spring Harbor perspectives in biology* 7.4, a005934.

Baas, Nils Andreas and Claus Emmeche (1997). "On emergence and explanation". In: *Intellectica* 25.2, pp. 67–83.

Bach, Sebastian et al. (2015). "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation". In: *PloS one* 10.7, e0130140.

Baker, Stephen et al. (2007). "A novel linear plasmid mediates flagellar variation in Salmonella Typhi". In: *PLoS Pathog* 3.5, e59.

Balcan, Duygu et al. (2007). "The information coded in the yeast response elements accounts for most of the topological properties of its transcriptional regulation network". In: *PLoS One* 2.6, e501.

Bansal, Mukesh, Giusy Della Gatta, and Diego Di Bernardo (2006). "Inference of gene regulatory networks and compound mode of action from time course gene expression profiles". In: *Bioinformatics* 22.7, pp. 815–822.

Bar-Joseph, Ziv et al. (2003). "Computational discovery of gene modules and regulatory networks". In: *Nature biotechnology* 21.11, pp. 1337–1342.

Barabási, Albert-László, Natali Gulbahce, and Joseph Loscalzo (2011). "Network medicine: a network-based approach to human disease". In: *Nature reviews genetics* 12.1, pp. 56–68.

Basso, Andrea D et al. (2002). "Akt forms an intracellular complex with heat shock protein 90 (Hsp90) and Cdc37 and is destabilized by inhibitors of Hsp90 function". In: *Journal of Biological Chemistry* 277.42, pp. 39858–39866.

Batagelj, Vladimir and Matjaz Zaversnik (2003). "An O (m) algorithm for cores decomposition of networks". In: *arXiv preprint cs/0310049*.

Bell, Graham and Arne O Mooers (1997). "Size and complexity among multicellular organisms". In: *Biological Journal of the Linnean Society* 60.3, pp. 345–363.

Benjamini, Yoav and Yosef Hochberg (1995). "Controlling the false discovery rate: a practical and powerful approach to multiple testing". In: *Journal of the Royal statistical society: series B (Methodological)* 57.1, pp. 289–300.

Biran, Or and Courtenay Cotton (2017). "Explanation and justification in machine learning: A survey". In: *IJCAI-17 workshop on explainable AI (XAI)*. Vol. 8. 1, pp. 8–13.

Bolouri, Hamid (2008). *Computational modeling of gene regulatory networks-a primer*. World Scientific Publishing Company.

Bonneau, Richard et al. (2006). "The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo". In: *Genome biology* 7.5, pp. 1–16.

Borgatti, Stephen P (2005). "Centrality and network flow". In: *Social networks* 27.1, pp. 55–71.

Boschetti, Fabio et al. (2005). "Defining and detecting emergence in complex networks". In: *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, pp. 573–580.

Boser, Bernhard E, Isabelle M Guyon, and Vladimir N Vapnik (1992). "A training algorithm for optimal margin classifiers". In: *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152.

Brandes, Ulrik (2001). "A faster algorithm for betweenness centrality". In: *Journal of mathematical sociology* 25.2, pp. 163–177.

— (2008). "On variants of shortest-path betweenness centrality and their generic computation". In: *Social networks* 30.2, pp. 136–145.

Brands, Roman C et al. (2017). "Targeting VEGFR and FGFR in head and neck squamous cell carcinoma in vitro". In: *Oncology reports* 38.3, pp. 1877–1885.

Breiman, Leo (1996). "Bagging predictors". In: *Machine learning* 24.2, pp. 123–140.

— (2001). "Random forests". In: *Machine learning* 45.1, pp. 5–32.

Broder, Andrei et al. (2011). "Graph structure in the web". In: *The Structure and Dynamics of Networks*. Princeton University Press, pp. 183–194.

Brolih, Sanja et al. (2018). "AKT1 restricts the invasive capacity of head and neck carcinoma cells harboring a constitutively active PI3 kinase activity". In: *BMC cancer* 18.1, pp. 1–10.

Cai, Gengming et al. (2020). "Identifying 8-mRNAsi based signature for predicting survival in patients with head and neck squamous cell carcinoma via machine learning". In: *Frontiers in genetics* 11, p. 1296.

Campbell, Kirsteen J et al. (2018). "MCL-1 is a prognostic indicator and drug target in breast cancer". In: *Cell death & disease* 9.2, pp. 1–14.

Cao, Zhixing and Ramon Grima (2018). "Linear mapping approximation of gene regulatory networks with stochastic dynamics". In: *Nature communications* 9.1, pp. 1–15.

Carlson, Jean M and John Doyle (2002). "Complexity and robustness". In: *Proceedings of the national academy of sciences* 99.suppl 1, pp. 2538–2545.

Chandran, Uma R et al. (2007). "Gene expression profiles of prostate cancer reveal involvement of multiple molecular pathways in the metastatic process". In: *BMC cancer* 7.1, pp. 1–21.

Chapfuwa, Paidamoyo et al. (2018). "Adversarial time-to-event modeling". In: *International Conference on Machine Learning*. PMLR, pp. 735–744.

Chen, Yen-Chen, Wan-Chi Ke, and Hung-Wen Chiu (2014). "Risk classification of cancer survival using ANN with gene expression data from multiple laboratories". In: *Computers in biology and medicine* 48, pp. 1–7.

Cheng, Qing et al. (2012). "Amplification and high-level expression of heat shock protein 90 marks aggressive phenotypes of human epidermal growth factor receptor 2 negative breast cancer". In: *Breast cancer research* 14.2, pp. 1–15.

Cheng, Wei-Yi, Tai-Hsien Ou Yang, and Dimitris Anastassiou (2013). "Development of a prognostic model for breast cancer survival in an open challenge environment". In: *Science translational medicine* 5.181, 181ra50–181ra50.

Cortes, Corinna and Vladimir Vapnik (1995). "Support-vector networks". In: *Machine learning* 20.3, pp. 273–297.

Cox, David R (1972). "Regression models and life-tables". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 34.2, pp. 187–202.

Csermely, Peter et al. (2013). "Structure and dynamics of core/periphery networks". In: *Journal of Complex Networks* 1.2, pp. 93–123.

Csete, Marie and John Doyle (2004). "Bow ties, metabolism and disease". In: *TRENDS in Biotechnology* 22.9, pp. 446–450.

Daniels, Bryan C et al. (2018). "Criticality distinguishes the ensemble of biological regulatory networks". In: *Physical review letters* 121.13, p. 138102.

Danielsen, Stine Aske et al. (2015). "Portrait of the PI3K/AKT pathway in colorectal cancer". In: *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer* 1855.1, pp. 104–121.

Ud-Dean, SM Minhaz et al. (2016). "TRaCE+: ensemble inference of gene regulatory networks from transcriptional expression profiles of gene knock-out experiments". In: *BMC bioinformatics* 17.1, pp. 1–14.

Diehl, Alexander D et al. (2016). "The Cell Ontology 2016: enhanced content, modularization, and ontology interoperability". In: *Journal of biomedical semantics* 7.1, pp. 1–10.

Dong, Yuan-Lin et al. (2013). "CXCR2-driven ovarian cancer progression involves upregulation of proinflammatory chemokines by potentiating NF-κB activation via EGFR-transactivated Akt signaling". In: *PloS one* 8.12, e83789.

Drucker, Harris et al. (1996). "Support vector regression machines". In: *Advances in neural information processing systems* 9.

Elmarakeby, Haitham A et al. (2021). "Biologically informed deep neural network for prostate cancer discovery". In: *Nature* 598.7880, pp. 348–352.

Faith, Jeremiah J et al. (2007). "Many Microbe Microarrays Database: uniformly normalized Affymetrix compendia with structured experimental metadata". In: *Nucleic acids research* 36.suppl_1, pp. D866–D870.

Farabaugh, Susan M, David N Boone, and Adrian V Lee (2015). "Role of IGF1R in breast cancer subtypes, stemness, and lineage differentiation". In: *Frontiers in endocrinology* 6, p. 59.

Ferreira, Gustavo Rodrigues, Helder Imoto Nakaya, and Luciano da Fontoura Costa (2018). "Gene regulatory and signaling networks exhibit distinct topological distributions of motifs". In: *Physical Review E* 97.4, p. 042417. DOI: https://doi.org/10.1103/PhysRevE.97.042417.

Fortelny, Nikolaus and Christoph Bock (2020). "Knowledge-primed neural networks enable biologically interpretable deep learning on single-cell sequencing data". In: *Genome biology* 21.1, pp. 1–36.

Freeman, Linton C, Douglas Roeder, and Robert R Mulholland (1979). "Centrality in social networks: II. Experimental results". In: *Social networks* 2.2, pp. 119–141.

Freund, Yoav and Robert E Schapire (1997). "A decision-theoretic generalization of on-line learning and an application to boosting". In: *Journal of computer and system sciences* 55.1, pp. 119–139.

Friedlander, Tamar et al. (2015). "Evolution of bow-tie architectures in biology". In: *PLoS computational biology* 11.3, e1004055.

Gama-Castro, Socorro et al. (2008). "RegulonDB (version 6.0): gene regulation model of Escherichia coli K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation". In: *Nucleic acids research* 36.suppl_1, pp. D120–D124.

Gardner, Timothy S, Diego Di Bernardo, et al. (2003). "Inferring genetic networks and identifying compound mode of action via expression profiling". In: *Science* 301.5629, pp. 102–105.

Gardner, Timothy S and Jeremiah J Faith (2005). "Reverse-engineering transcription control networks". In: *Physics of life reviews* 2.1, pp. 65–88.

Gáspár, Merse E and Peter Csermely (2012). "Rigidity and flexibility of biological networks". In: *Briefings in Functional Genomics* 11.6, pp. 443–456.

Gerstein, Mark B et al. (2012). "Architecture of the human regulatory network derived from ENCODE data". In: *Nature* 489.7414, pp. 91–100.

Ghosh Roy, Gourab, Nicholas Geard, et al. (2020). "PoLoBag: Polynomial Lasso Bagging for signed gene regulatory network inference from expression data". In: *Bioinformatics* 36.21, pp. 5187–5193.

Ghosh Roy, Gourab, Shan He, et al. (2021). "Bow-tie architecture of gene regulatory networks in species of varying complexity". In: *Journal of the Royal Society Interface* 18.179, p. 20210069.

Giatsidis, Christos et al. (2014). "Quantifying trust dynamics in signed graphs, the S-Cores approach". In: *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, pp. 668–676.

Goldman, Mary J et al. (2020). "Visualizing and interpreting cancer genomics data via the Xena platform". In: *Nature biotechnology* 38.6, pp. 675–678.

Griss, Johannes et al. (2020). "Reactomegsa-efficient multi-omics comparative pathway analysis". In: *Molecular & Cellular Proteomics* 19.12, pp. 2115–2125.

Gupta, Rita et al. (2011). "A computational framework for gene regulatory network inference that combines multiple methods and datasets". In: *BMC systems biology* 5.1, pp. 1–14.

Gustafsson, Mika et al. (2009). "Reverse engineering of gene networks with LASSO and nonlinear basis functions". In: *Challenges of Systems Biology: Community Efforts to Harness Biological Complexity* 1158, pp. 265–275.

Han, Heonjong et al. (2018). "TRRUST v2: an expanded reference database of human and mouse transcriptional regulatory interactions". In: *Nucleic acids research* 46.D1, pp. D380–D386.

Hanahan, Douglas and Robert A Weinberg (2011). "Hallmarks of cancer: the next generation". In: *cell* 144.5, pp. 646–674.

Harrell Jr, Frank E, Kerry L Lee, Robert M Califf, et al. (1984). "Regression modelling strategies for improved prognostic prediction". In: *Statistics in medicine* 3.2, pp. 143–152.

Harrell Jr, Frank E, Kerry L Lee, and Daniel B Mark (1996). "Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors". In: *Statistics in medicine* 15.4, pp. 361–387.

Hashimoto, M et al. (2013). "Protooncogene TCL1b functions as an Akt kinase co-activator that exhibits oncogenic potency in vivo". In: *Oncogenesis* 2.9, e70–e70.

Hatleberg, William L and Veronica F Hinman (2021). "Modularity and hierarchy in biological systems: Using gene regulatory networks to understand evolutionary change". In: *Current topics in developmental biology*. Vol. 141. Elsevier, pp. 39–73.

Hayes, Josie et al. (2015). "Prediction of clinical outcome in glioblastoma using a biologically relevant nine-microRNA signature". In: *Molecular oncology* 9.3, pp. 704–714.

Hedges, S Blair et al. (2004). "A molecular timescale of eukaryote evolution and the rise of complex multicellular life". In: *BMC evolutionary biology* 4.1, pp. 1–9.

Hermeking, Heiko et al. (2000). "Identification of CDK4 as a target of c-MYC". In: *Proceedings of the National Academy of Sciences* 97.5, pp. 2229–2234.

Hilten, Arno van et al. (2020). "GenNet framework: interpretable neural networks for phenotype prediction". In: *bioRxiv*.

Hinton, Geoffrey E and Ruslan R Salakhutdinov (2006). "Reducing the dimensionality of data with neural networks". In: *science* 313.5786, pp. 504–507.

Hinz, Nico and Manfred Jücker (2019). "Distinct functions of AKT isoforms in breast cancer: a comprehensive review". In: *Cell Communication and Signaling* 17.1, pp. 1–29.

Huang, Sui, Ingemar Ernberg, and Stuart Kauffman (2009). "Cancer attractors: a systems view of tumors from a gene network dynamics and developmental perspective". In: *Seminars in cell & developmental biology*. Vol. 20. 7. Elsevier, pp. 869–876.

Huang, Zhi et al. (2019). "SALMON: survival analysis learning with multi-omics neural networks on breast cancer". In: *Frontiers in genetics* 10, p. 166.

Huynh-Thu, Vân Anh and Pierre Geurts (2018). "dynGENIE3: dynamical GENIE3 for the inference of gene networks from time series expression data". In: *Scientific reports* 8.1, pp. 1–12.

Huynh-Thu, Vân Anh, Alexandre Irrthum, et al. (2010). "Inferring regulatory networks from expression data using tree-based methods". In: *PloS one* 5.9, e12776.

Jiang, Ningni et al. (2020). "Role of PI3K/AKT pathway in cancer: the framework of malignant behavior". In: *Molecular biology reports* 47.6, pp. 4587–4629.

Jin, Ting et al. (2021). "ECMarker: Interpretable machine learning model identifies gene expression biomarkers predicting clinical outcomes and reveals molecular mechanisms of human disease in early stages". In: *Bioinformatics* 37.8, pp. 1115–1124.

Jothi, Raja et al. (2009). "Genomic analysis reveals a tight link between transcription factor dynamics and regulatory network architecture". In: *Molecular systems biology* 5.1, p. 294.

Kanehisa, Minoru, Miho Furumichi, et al. (2021). "KEGG: integrating viruses and cellular organisms". In: *Nucleic acids research* 49.D1, pp. D545–D551.

Kanehisa, Minoru and Susumu Goto (2000). "KEGG: kyoto encyclopedia of genes and genomes". In: *Nucleic acids research* 28.1, pp. 27–30.

Karlebach, Guy and Ron Shamir (2008). "Modelling and analysis of gene regulatory networks". In: *Nature reviews Molecular cell biology* 9.10, pp. 770–780.

Katzman, Jared L et al. (2018). "DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network". In: *BMC medical research methodology* 18.1, pp. 1–12.

Kauffman, Stuart (1969). "Homeostasis and differentiation in random genetic control networks". In: *Nature* 224.5215, pp. 177–178.

Kauffman, Stuart A (1969). "Metabolic stability and epigenesis in randomly constructed genetic nets". In: *Journal of theoretical biology* 22.3, pp. 437–467.

Khosravi, Pegah et al. (2015). "Inferring interaction type in gene regulatory networks using co-expression data". In: *Algorithms for molecular biology* 10.1, pp. 1–11.

Kim, Jeong-Rae, Yeoin Yoon, and Kwang-Hyun Cho (2008). "Coupled feedback loops form dynamic motifs of cellular networks". In: *Biophysical journal* 94.2, pp. 359–365.

Kim, Jongkwang and Thomas Wilhelm (2008). "What is a complex graph?" In: *Physica A: Statistical Mechanics and its Applications* 387.11, pp. 2637–2652.

King, Mary-Claire and Allan C Wilson (1975). "Evolution at two levels in humans and chimpanzees". In: *Science* 188.4184, pp. 107–116.

Kirschner, Marc and John Gerhart (1998). "Evolvability". In: *Proceedings of the National Academy of Sciences* 95.15, pp. 8420–8427.

Kitano, Hiroaki (2004a). "Biological robustness". In: *Nature Reviews Genetics* 5.11, pp. 826–837.

— (2004b). "Cancer as a robust system: implications for anticancer therapy". In: *Nature Reviews Cancer* 4.3, pp. 227–235.

Kitsak, Maksim et al. (2010). "Identification of influential spreaders in complex networks". In: *Nature physics* 6.11, p. 888.

Klahan, Sukhontip et al. (2014). "Computational analysis of mRNA expression profiles identifies the ITG family and PIK3R3 as crucial genes for regulating triple negative breast cancer cell migration". In: *BioMed research international* 2014.

Kubí, Aleš (2003). "Toward a formalization of emergence". In: *Artificial life* 9.1, pp. 41–65.

Kuenzi, Brent M et al. (2020). "Predicting drug response and synergy using a deep learning model of human cancer cells". In: *Cancer cell* 38.5, pp. 672–684.

Küffner, Robert et al. (2010). "Petri nets with fuzzy logic (PNFL): reverse engineering and parametrization". In: *PLoS One* 5.9, e12807.

Kumar, Santhust et al. (2015). "Analysis of the hierarchical structure of the B. subtilis transcriptional regulatory network". In: *Molecular BioSystems* 11.3, pp. 930–941.

Kurian, Allison W, Bronislava M Sigal, and Sylvia K Plevritis (2010). "Survival analysis of cancer risk reduction strategies for BRCA1/2 mutation carriers". In: *Journal of Clinical Oncology* 28.2, p. 222.

Lambert, John, Ozan Sener, and Silvio Savarese (2018). "Deep learning under privileged information using heteroscedastic dropout". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8886–8895.

Laughlin, Robert B and David Pines (2000). "From the cover: The theory of everything". In: *Proceedings of the national academy of sciences of the United States of America* 97.1, p. 28.

Lee, Wei-Po and Wen-Shyong Tzou (2009). "Computational methods for discovering gene networks from expression data". In: *Briefings in bioinformatics* 10.4, pp. 408–423.

Licausi, Francesco et al. (2011). "HRE-type genes are regulated by growth-related changes in internal oxygen concentrations during the normal development of potato (Solanum tuberosum) tubers". In: *Plant and Cell Physiology* 52.11, pp. 1957–1972.

Lin, Chieh et al. (2017). "Using neural networks for reducing the dimensions of single-cell RNA-Seq data". In: *Nucleic acids research* 45.17, e156–e156.

Liu, Jianfang et al. (2018). "An integrated TCGA pan-cancer clinical data resource to drive high-quality survival outcome analytics". In: *Cell* 173.2, pp. 400–416.

Liu, Zhi-Ping et al. (2015). "RegNetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse". In: *Database* 2015.

Llovet, Josep M et al. (2012). "Plasma biomarkers as predictors of outcome in patients with advanced hepatocellular carcinoma". In: *Clinical Cancer Research* 18.8, pp. 2290–2300.

Long, Terri A, Siobhan M Brady, and Philip N Benfey (2008). "Systems approaches to identifying gene regulatory networks in plants". In: *Annual review of cell and developmental biology* 24, pp. 81–103.

Lü, Jinhu et al. (2013). "Theory and applications of complex networks: Advances and challenges". In: *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, pp. 2291–2294.

Luo, Shitao et al. (2018). "Similar bowtie structures and distinct largest strong components are identified in the transcriptional regulatory networks of Arabidopsis thaliana during photomorphogenesis and heat shock". In: *Biosystems* 168, pp. 1–7.

Ma, Hong-Wu and An-Ping Zeng (2003). "The connectivity structure, giant strong component and centrality of metabolic networks". In: *Bioinformatics* 19.11, pp. 1423–1430.

Ma, Jianzhu et al. (2018). "Using deep learning to model the hierarchical structure and function of a cell". In: *Nature methods* 15.4, p. 290.

Ma, Tianle and Aidong Zhang (2018). "Multi-view factorization AutoEncoder with network constraints for multi-omic integrative analysis". In: *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, pp. 702–707.

Malkoç, Berkin, Duygu Balcan, and Ayşe Erzan (2010). "Information content based model for the topological properties of the gene regulatory network of Escherichia coli". In: *Journal of theoretical biology* 263.3, pp. 281–294.

Mann, Henry B and Donald R Whitney (1947). "On a test of whether one of two random variables is stochastically larger than the other". In: *The annals of mathematical statistics*, pp. 50–60.

Marbach, Daniel, Robert J Prill, et al. (2010). "Revealing strengths and weaknesses of methods for gene network inference". In: *Proceedings of the national academy of sciences* 107.14, pp. 6286–6291.

Marbach, Daniel, Thomas Schaffter, Dario Floreano, et al. (2009). "The DREAM4 in-silico network challenge". In: *Draft, version 0.3*.

Marbach, Daniel, Thomas Schaffter, Claudio Mattiussi, et al. (2009). "Generating realistic in silico gene networks for performance assessment of reverse engineering methods". In: *Journal of computational biology* 16.2, pp. 229–239.

Margolin, Adam A et al. (2006). "ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context". In: *BMC bioinformatics*. Vol. 7. 1. Springer, pp. 1–15.

Marion, Zachary H, James A Fordyce, and Benjamin M Fitzpatrick (2015). "Extending the concept of diversity partitioning to characterize phenotypic complexity". In: *The American Naturalist* 186.3, pp. 348–361.

Marisi, Giorgia et al. (2018). "Ten years of sorafenib in hepatocellular carcinoma: Are there any predictive and/or prognostic markers?" In: *World journal of gastroenterology* 24.36, p. 4152.

Mason, Mike J et al. (2009). "Signed weighted gene co-expression network analysis of transcriptional regulation in murine embryonic stem cells". In: *BMC genomics* 10.1, pp. 1–25.

McCulloch, Warren S and Walter Pitts (1943). "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4, pp. 115–133.

Middendorf, Manuel et al. (2004). "Predicting genetic regulatory response using classification". In: *Bioinformatics* 20.suppl_1, pp. i232–i240.

Minami, Takashi et al. (2013). "The calcineurin-NFAT-angiopoietin-2 signaling axis in lung endothelium is critical for the establishment of lung metastases". In: *Cell reports* 4.4, pp. 709–723.

Montojo, Jason et al. (2015). "SIREN Cytoscape plugin: interaction type discrimination in gene regulatory networks". In: *arXiv preprint arXiv:1512.05067*.

Morgan, Daniel et al. (2019). "A generalized framework for controlling FDR in gene regulatory network inference". In: *Bioinformatics* 35.6, pp. 1026–1032.

Muhammad, Durreshahwar et al. (2017). "More than meets the eye: Emergent properties of transcription factors networks in Arabidopsis". In: *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms* 1860.1, pp. 64–74.

Murali, Thilakam et al. (2011). "DroID 2011: a comprehensive, integrated resource for protein, transcription factor, RNA and gene interactions for Drosophila". In: *Nucleic acids research* 39.suppl_1, pp. D736–D743.

Myers, Andrea P and Lewis C Cantley (2010). "Targeting a common collaborator in cancer development". In: *Science translational medicine* 2.48, 48ps45–48ps45.

Narang, Vipin et al. (2015). "Automated identification of core regulatory genes in human gene regulatory networks". In: *PLoS computational biology* 11.9, e1004504.

Natarajan, Anirudh et al. (2012). "Predicting cell-type–specific gene expression from regions of open chromatin". In: *Genome research* 22.9, pp. 1711–1722.

Nurse, Paul (2008). "Life, logic and information". In: *Nature* 454.7203, pp. 424–426.

Oh, Jung Hun et al. (2021). "PathCNN: interpretable convolutional neural networks for survival prediction and pathway analysis applied to glioblastoma". In: *Bioinformatics* 37.Supplement_1, pp. i443–i450.

Ouma, Wilberforce Zachary, Katja Pogacar, and Erich Grotewold (2018). "Topological and statistical analyses of gene regulatory networks reveal unifying yet quantitatively different emergent properties". In: *PLoS computational biology* 14.4, e1006098.

Page, Lawrence et al. (1999). *The PageRank citation ranking: Bringing order to the web.* Tech. rep. Stanford InfoLab.

Parise, Mariana Teixeira Dornelles et al. (2020). "CoryneRegNet 7, the reference database and analysis platform for corynebacterial gene regulatory networks". In: *Scientific data* 7.1, pp. 1–9.

Paul, Torsten Johann and Philip Kollmannsberger (2020). "Biological network growth in complex environments: A computational framework". In: *PLOS Computational Biology* 16.11, e1008003.

Peixoto, Tiago P (2012). "Emergence of robustness against noise: A structural phase transition in evolved models of gene regulatory networks". In: *Physical Review E* 85.4, p. 041908.

Pellagatti, A et al. (2010). "Deregulated gene expression pathways in myelodysplastic syndrome hematopoietic stem cells". In: *Leukemia* 24.4, pp. 756–764.

Prill, Robert J et al. (2010). "Towards a rigorous assessment of systems biology models: the DREAM3 challenges". In: *PloS one* 5.2, e9202.

Rodriguez-Caso, Carlos, Bernat Corominas-Murtra, and Ricard V Solé (2009). "On the basic computational structure of gene regulatory networks". In: *Molecular BioSystems* 5.12, pp. 1617–1629.

Roli, Andrea et al. (2018). "Dynamical criticality: overview and open questions". In: *Journal of Systems Science and Complexity* 31.3, pp. 647–663.

Roy, Gourab Ghosh et al. (2022). *MPVNN: Mutated Pathway Visible Neural Network Architecture for Interpretable Prediction of Cancer-specific Survival Risk*. arXiv: 2202.00882 [q-bio.QM].

Rudin, Cynthia (2019). "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". In: *Nature Machine Intelligence* 1.5, pp. 206–215.

Sakamoto, Kathleen M and David A Frank (2009). "CREB in the pathophysiology of cancer: implications for targeting transcription factors for cancer therapy". In: *Clinical Cancer Research* 15.8, pp. 2583–2587.

Saleh, Mahmoud, Yusef Esa, and Ahmed Mohamed (2018). "Applications of complex network analysis in electric power systems". In: *Energies* 11.6, p. 1381.

Sanders, Laura E and Blake Cady (1998). "Differentiated thyroid cancer: reexamination of risk groups and outcome of treatment". In: *Archives of Surgery* 133.4, pp. 419–425.

Santos-Zavaleta, Alberto et al. (2019). "RegulonDB v 10.5: tackling challenges to unify classic and high throughput knowledge of gene regulation in E. coli K-12". In: *Nucleic acids research* 47.D1, pp. D212–D220.

Sarris, Evangelos G, Muhammad W Saif, and Kostas N Syrigos (2012). "The biological role of PI3K pathway in lung cancer". In: *Pharmaceuticals* 5.11, pp. 1236–1264.

Schaffter, Thomas, Daniel Marbach, and Dario Floreano (2011). "GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods". In: *Bioinformatics* 27.16, pp. 2263–2270.

Seidman, Stephen B (1983). "Network structure and minimum degree". In: *Social networks* 5.3, pp. 269–287.

Shalizi, Cosma Rohilla, Kristina Lisa Shalizi, and James P Crutchfield (2002). "An algorithm for pattern discovery in time series". In: *arXiv preprint cs/0210025*.

Shrikumar, Avanti, Peyton Greenside, and Anshul Kundaje (2017). "Learning important features through propagating activation differences". In: *International conference on machine learning*. PMLR, pp. 3145–3153.

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman (2013). "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint arXiv:1312.6034*.

Smythe, W Roy et al. (2001). "Surgical resection of non–small cell carcinoma after treatment for small cell carcinoma". In: *The Annals of thoracic surgery* 71.3, pp. 962–966.

Snel, Berend et al. (2000). "STRING: a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene". In: *Nucleic acids research* 28.18, pp. 3442–3444.

Son, Deok-Soo et al. (2013). "Characteristics of chemokine signatures elicited by EGF and TNF in ovarian cancer cells". In: *Journal of inflammation* 10.1, pp. 1–12.

Steck, Harald et al. (2008). "On ranking in survival analysis: Bounds on the concordance index". In: *Advances in neural information processing systems*, pp. 1209–1216.

Sun, Dongdong, Minghui Wang, and Ao Li (2018). "A multimodal deep neural network for human breast cancer prognosis prediction by integrating multi-dimensional data". In: *IEEE/ACM transactions on computational biology and bioinformatics* 16.3, pp. 841–850.

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan (2016). "Gradients of counterfactuals". In: *arXiv preprint arXiv:1611.02639*.

Supper, Jochen et al. (2009). "BowTieBuilder: modeling signal transduction pathways". In: *BMC systems biology* 3.1, pp. 1–13.

Tan, Jie et al. (2014). "Unsupervised feature construction and knowledge extraction from genome-wide assays of breast cancer with denoising autoencoders". In: *Pacific symposium on biocomputing co-chairs*. World Scientific, pp. 132–143.

Tavassoly, Iman, Joseph Goldfarb, and Ravi Iyengar (2018). "Systems biology primer: the basic methods and approaches". In: *Essays in biochemistry* 62.4, pp. 487–500.

Tibshirani, Robert (1996). "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1, pp. 267–288.

Tieri, Paolo et al. (2010). "Network, degeneracy and bow tie. Integrating paradigms and architectures to grasp the complexity of the immune system". In: *Theoretical Biology and Medical Modelling* 7.1, pp. 1–16.

Tomczak, Katarzyna, Patrycja Czerwińska, and Maciej Wiznerowicz (2015). "The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge". In: *Contemporary oncology* 19.1A, A68.

Torrente, Aurora et al. (2016). "Identification of cancer related genes using a comprehensive map of human gene expression". In: *PloS one* 11.6, e0157484.

Torres-Sosa, Christian, Sui Huang, and Maximino Aldana (2012). "Criticality is an emergent property of genetic networks that exhibit evolvability". In.

Tyson, John J, Katherine C Chen, and Bela Novak (2003). "Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell". In: *Current opinion in cell biology* 15.2, pp. 221–231.

Uzunangelov, Vladislav, Christopher K Wong, and Joshua M Stuart (2021). "Accurate cancer phenotype prediction with AKLIMATE, a stacked kernel learner integrating multimodal genomic data and pathway knowledge". In: *PLoS Computational Biology* 17.4, e1008878.

Vafaee, Fatemeh et al. (2016). "ORTI: an open-access repository of transcriptional interactions for interrogating mammalian gene expression data". In: *PloS one* 11.10, e0164535.

Veber, Philippe et al. (2008). "Inferring the role of transcription factors in regulatory networks". In: *BMC bioinformatics* 9.1, pp. 1–21.

Veličković, Petar et al. (2017). "Graph attention networks". In: *arXiv preprint arXiv:1710.10903*.

Vickaryous, Matthew K and Brian K Hall (2006). "Human cell type diversity, evolution, development, and classification with special reference to cells derived from the neural crest". In: *Biological reviews* 81.3, pp. 425–455.

Wang, Sheng et al. (2021). "Leveraging the Cell Ontology to classify unseen cell types". In: *Nature communications* 12.1, pp. 1–11.

Wang, Sijian et al. (2011). "Random lasso". In: *The annals of applied statistics* 5.1, p. 468.

Wang, Le-Zhi et al. (2016). "A geometrical approach to control and controllability of nonlinear dynamical networks". In: *Nature communications* 7.1, pp. 1–11.

Weng, Gezhi, Upinder S Bhalla, and Ravi Iyengar (1999). "Complexity in biological signaling systems". In: *Science* 284.5411, pp. 92–96.

Wetterstrand, K A (2016). *DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP)*. URL: www.genome.gov/sequencingcosts.

Wilcoxon, Frank (1992). "Individual comparisons by ranking methods". In: *Breakthroughs in statistics*. Springer, pp. 196–202.

Wittkopp, Patricia J (2007). "Variable gene expression in eukaryotes: a network perspective". In: *Journal of Experimental Biology* 210.9, pp. 1567–1575.

Wulczyn, Ellery et al. (2020). "Deep learning-based survival prediction for multiple cancer types using histopathology images". In: *PLoS One* 15.6, e0233678.

Xu, Xiaoyi et al. (2012). "A gene signature for breast cancer prognosis using support vector machine". In: *2012 5th International Conference on BioMedical Engineering and Informatics*. IEEE, pp. 928–931.

Yamada, Takuji, Minoru Kanehisa, and Susumu Goto (2006). "Extraction of phylogenetic network modules from the metabolic network". In: *BMC bioinformatics* 7.1, pp. 1–10.

Yang, Rong, Leyla Zhuhadar, and Olfa Nasraoui (2011). "Bow-tie decomposition in directed graphs". In: *14th International Conference on Information Fusion*. IEEE, pp. 1–5.

Yang, Tzu-Hsien et al. (2014). "YTRP: a repository for yeast transcriptional regulatory pathways". In: *Database* 2014.

Yilmaz, Alper et al. (2010). "AGRIS: the Arabidopsis gene regulatory information server, an update". In: *Nucleic acids research* 39.suppl_1, pp. D1118–D1122.

Yu, Haiyuan and Mark Gerstein (2006). "Genomic analysis of the hierarchical structure of regulatory networks". In: *Proceedings of the National Academy of Sciences* 103.40, pp. 14724–14731.

Yu, Jing et al. (2004). "Advances to Bayesian network inference for generating causal networks from observational biological data". In: *Bioinformatics* 20.18, pp. 3594–3603.

Yu, Michael K et al. (2018). "Visible machine learning for biomedicine". In: *Cell* 173.7, pp. 1562–1565.

Zak, Daniel E et al. (2003). "Importance of input perturbations and stochastic gene expression in the reverse engineering of genetic regulatory networks: insights from an identifiability analysis of an in silico network". In: *Genome research* 13.11, pp. 2396–2405.

Zanudo, Jorge GT and Réka Albert (2015). "Cell fate reprogramming by control of intracellular network dynamics". In: *PLoS computational biology* 11.4, e1004193.

Zhang, J and S Zhang (2013). "Modular Organization of Gene Regulatory Networks". In: *Encyclopedia of Systems Biology*, pp. 1437–1441.

Zou, Hui and Trevor Hastie (2005). "Regularization and variable selection via the elastic net". In: *Journal of the royal statistical society: series B (statistical methodology)* 67.2, pp. 301–320.