Virginia Commonwealth University
VCU Scholars Compass

Theses and Dissertations | Graduate School

2015

In-silico Models for Capturing the Static and Dynamic Characteristics of Robustness within Complex Networks

Bhanu K. Kamapantula

Follow this and additional works at: https://scholarscompass.vcu.edu/etd

Part of the Systems Biology Commons

© The Author

Downloaded from
https://scholarscompass.vcu.edu/etd/4049

This Dissertation is brought to you for free and open access by the Graduate School at VCU Scholars Compass. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of VCU Scholars Compass. For more information, please contact libcompass@vcu.edu.

©Bhanu K. Kamapantula, December 2015

Copyleft.

DISSERTATION

IN-SILICO MODELS FOR CAPTURING THE STATIC AND DYNAMIC CHARACTERISTICS OF ROBUSTNESS WITHIN COMPLEX NETWORKS

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

by

BHANU KAMAPANTULA

B.E., Osmania University - September 2004 to May 2008

Texas A&M University at Corpus Christi - Fall 2009 to Summer 2011

Director: Dissertation Dr. Preetam Ghosh,

Associate Professor, Department of Computer Science

Committee Members

Dr. Michael Mayo

Dr. Wei Cheng

Dr. Thang Dinh

Dr. Danail Bonchev

Dr. Kevin Pilkiewicz

Virginia Commonwealth University

Richmond, Virginia

December, 2015

Acknowledgements

I am thankful to the committee members: Dr. Wei Cheng, Dr. Thang Dinh, Dr. Danail Bonchev and Dr. Kevin Pilkiewicz, who agreed to serve on the committee. Special thanks to Dr. Preetam Ghosh, my advisor since Fall 2011, for his continuous support and to Dr. Michael Mayo, my collaborator, for his support.

I am grateful to all my parents, brother, other family members, many friends who have supported me during the course of my study. I am also thankful to a) the free and open source community that developed software I have been using for many years including Ubuntu and Python, b) the developers and community of StackOverFlow.com for the high quality discussions that are an information gold mine to me and numerous other online research resources that continue to help me grow as a researcher and developer.

TABLE OF CONTENTS

Chapter Page

Acknowledgements iii

Table of Contents iv

List of Tables vi

List of Figures vii

Abstract xiii

1 Introduction 1

1.1 Wireless Sensor Networks 1
1.2 Why *E. coli* and *Yeast*? 3

2 Contributions 5

3 Robustness 7

3.1 Metrics and definitions 8
3.1.1 Robustness after attacks 9
3.2 Biological Networks 10
3.3 Quantifying robustness in biological networks using NS-2 10
3.4 Computational modelling 11
3.5 Mapping GRNs to WSNs 13
3.6 NS-2 simulation setup 15
3.6.1 Network generation 17
3.6.2 Sink selection strategy 18
3.6.2.1 New network generation algorithm 20
3.6.3 SVM Validation 23
3.6.3.1 Network density (ND) 25
3.6.3.2 Genes coverage (GC) 25
3.6.3.3 Transcription factor network density (TND) 25
3.6.3.4 Motif abundance 25
3.6.3.5 Genes percentage (GP) 25
3.6.4 Contributions of topological metrics to GRN robustness . . . 25

3.7 Case study: Comparison of derived networks from E. coli and Yeast . . . 26
3.8 Challenges and Future directions . . . 29

4 Structural redundancy of transcriptional motifs . . . 34

4.1 Introduction . . . 34
4.2 Methods . . . 36
4.2.1 Model transcriptional networks . . . 36
4.2.2 Simulation setup . . . 36
4.2.3 Motif structural redundancy and packet receipt . . . 38
4.3 Support Vector Machine Modeling . . . 40
4.3.1 Assigning labels for SVM . . . 41
4.3.2 Data pruning . . . 42
4.3.3 Training and testing . . . 42
4.3.4 Parameter selection . . . 43
4.3.5 Features . . . 43
4.3.5.1 Network density . . . 45
4.3.5.2 Average shortest path . . . 45
4.3.5.3 Degree centrality . . . 45
4.3.5.4 Transcription factor percentage . . . 46
4.3.5.5 Genes percentage . . . 46
4.3.5.6 Source to sink edge percentage . . . 46
4.3.5.7 FFL abundance . . . 47
4.3.5.8 FFLDED . . . 47
4.3.5.9 FFLSSPD . . . 47
4.3.5.10 FFLDEP . . . 48
4.3.5.11 FFLIDEP . . . 48
4.3.5.12 Direct-edge trace participation . . . 49
4.3.5.13 Indirect-edge trace participation . . . 49
4.3.6 Feature ranking . . . 49
4.4 Results . . . 50
4.4.1 Packet receipt rates using transcriptional network topologies 50
4.4.2 Feature ranking in transcriptional networks . . . 50
4.4.2.1 Top-ranking features . . . 51
4.4.2.2 Feature stability at different perturbation levels . . . 51
4.4.2.3 Feature ranking variation across different network sizes . . . 52
4.4.2.4 Comparison of FFL based features . . . 52

4.5 Discussion and conclusions 53
4.6 Biorobust 54

5 Abundance of connected motifs in transcriptional networks 63

5.1 Introduction 63
5.2 Methodology 64
5.2.1 Contributions 65
5.2.2 Transcriptional subnetworks 65
5.2.3 Modeling network dynamics using NS-2 67
5.2.4 Structural features 68
5.2.5 Vertex-shared motif connectivity 69
5.3 Random forest regression 70
5.3.1 Data 71
5.3.2 Regression modeling 72
5.3.3 Feature reduction 75
5.3.4 Feature value correlation with robustness 76
5.4 Vertex-shared motifs 77
5.5 Discussion 79

6 Role of FFLs in signal transduction 87

6.1 Signal transduction 92

7 Conclusion 100

8 Future work 101

References 115

LIST OF TABLES

Table Page

1 Feedforward Loop Motif Count in RWSNs and new algorithm networks [26] 21

2 Attachment kernels 23

3 Grid search parameters identified using the cross validation method described in the text (20% perturbation). 43

4 Abundance of bow-tie and bi-triangle motifs in *E. coli* transcriptional network [61]. 84

5 Abundance of rhombus motifs in *E. coli* transcriptional network [61]. . . 85

6 Isomorphic rhombus motifs in *E. coli* transcriptional network [61]. 86

7 Mean and standard deviation (STD) of features - 100 network size at under different perturbation levels. 99

8 Mean and standard deviation (STD) of features - 200 network size at under different perturbation levels. 111

9 Mean and standard deviation (STD) of features - 300 network size at under different perturbation levels. 112

10 Mean and standard deviation (STD) of features - 400 network size at under different perturbation levels. 113

11 Mean and standard deviation (STD) of features - 500 network size at under different perturbation levels. 114

LIST OF FIGURES

Figure Page

1 Sink node selection and respective packet receival rates for different loss models - GRN of 20 nodes [30] 18

2 Comparison of best, mean and worst (out of 25 networks) performing RWSNs to GRN - Network size 100 [26] 20

3 Comparison of best, mean and worst (out of 25 networks) performing RWSNs to GRN - Network size 300 [26] 21

4 Comparison of average of ten networks between New Algorithm and RWSN - Network size 1500 24

5 Comparison of average of ten networks between New Algorithm and RWSN - Network size 2500 24

6 (a) Relative importance of the feature weights, (b) Relative importance of feature directions 26

7 Comparison of best performing networks derived from E. coli and Yeast - 20%, 35% and 50% loss 30

8 Comparison of mean performing networks derived from E. coli and Yeast - 20%, 35% and 50% loss 31

9 Comparison of worst performing networks derived from E. coli and Yeast - 20%, 35% and 50% loss 32

10 Comparison of 100 node networks derived from E.coli and Yeast respectively - 20%, 35%, 50% loss 33

11 Comparison of 500 node networks derived from E.coli and Yeast respectively - 20%, 35%, 50% loss 33

12 Procedure followed to identify significant features in *E. coli* and *Yeast* subnetworks 37

13 Packet receipt rates (PRTs) for sampled transcriptional subnetworks of the bacterium *Escherichia coli* and *Saccharomyces cerevisiae* (labeled 'Yeast'). 39

14 Illustration of (a) SVM Dataset for each network size, at specific perturbation level and (b)(1) FFL, (b)(2) bifan motifs respectively 44

15 Variation of top 5 features in each *Escherichia coli* network (panel (a)) and *Saccharomyces cerevisiae* (panel (b)) networks, at losses 20% and 50% (Sizes = 100, 200, 300, 400, 500). 56

16 Variation in normalized ANOVA F-values for the top 5 features in each *Escherichia coli* network (panel (a)) and *Saccharomyces cerevisiae* (panel (b)) networks, at losses 20% and 50% (Sizes = 100, 200, 300, 400, 500). 57

17 Variation of FFL participating direct and indirect edge-based features at 20%, 35% and 50% loss each for different E. coli networks (Sizes = 100, 200, 300, 400, 500). 58

18 Variation of FFL participating direct and indirect edge-based features at 20%, 35% and 50% loss each for different Yeast networks (Sizes = 100, 200, 300, 400, 500). 59

19 BioRobust flowchart 60

20 BioRobust prototype - Select network for analysis 60

21 BioRobust prototype - User notification of results (Email blurred). 61

22 BioRobust prototype - Robustness analysis of the uploaded network across different perturbation levels. 61

23 BioRobust prototype - Feature importance of the uploaded network across different perturbation levels. 62

24 (a) Step-by-step methodology (b) FFL motif 66

25 Mean square errors (MSE) at different estimators for 400 network size at 90% loss. Measured for the model with 38 features. Errorbars capture the variation of MSE across hundred test runs. Note that the Y-axis does not start at 0. *Lower MSE is better.* 74

26 Coefficient of determination (COD) for regressors - different network sizes for 38 features model. Each data point represents an average value across 100 runs. *Higher COD is better* [61]. 78

27 (a) Selected features (out of total 38) for every model at a given network size and loss model as described in Section 5.3.2. (b) Selected features (out of 23) feed-forward loop connected motifs. Each data point represents an average value across 100 runs. Criteria: select features that have higher than average feature importance using random forest regression [61]. 80

28 (a) Feature significance in all the networks at 60% loss for model with all 38 features. (b) Feature significance of connected feed-forward loop motifs in all the networks at loss 60%. *The darker the color the higher the feature significance.* Additionally, numbers are included to indicate feature rank. Each data point represents an average value across 100 runs. *Higher the feature importance, better is the feature* [61]. 81

29 Feature importances as determined by random forest regression for models with 38 features and 23 features respectively for network size 100 at 75% loss. While the feature source to sink edge percentage is termed as SSEP, the percentage of FFL direct edges is termed FFLDEP. The features 10, 11 as explained in Section 5.2.4 are FFLIDSPATH and FFLDOSPATH. Refer to Table 4 and Table 5 for definitions of BW-1, BW-2, BW-4, and RH-7. *Higher feature importance is better.* . . . 83

30 Feature importance - 100 network size. SSEP feature stands out with increase in noise. Each of the feature value mentioned in the heat map cell is an average across 100 runs. 88

31 Feature importance - 200 network size. While network density and average degree centrality (ADC) stand out upto loss 35%; SSEP, FFLIDEP features stand out with increase in noise. Each of the feature value mentioned in the heat map cell is an average across 100 runs. 89

32 Feature importance - 300 network size. The percentage of transcription factor and gene nodes are important at higher noise. FFLIDEP feature stands out with increase in noise. Each of the feature value mentioned in the heat map cell is an average across 100 runs. 90

33 Feature importance - 400 network size. FFLIDEP feature stands out with increase in noise. Each of the feature value mentioned in the heat map cell is an average across 100 runs. 91

34 Feature importance - 500 network size. FFLIDEP along with other FFL-derived features stand out with increase in noise. Each of the feature value mentioned in the heat map cell is an average across 100 runs. 92

35 Distribution of scaled feature importance values - 500 network size at 20% perturbation level. Each feature is measured across 100 runs. 93

36 Distribution of scaled feature importance values - 500 network size at 50% perturbation level. Each feature is measured across 100 runs. 94

37 Relative feature importance of FFLDEP vs FFLIDEP - all network sizes at different perturbation levels. Each data point is an average of 100 runs. 95

38 Distribution of coefficient of determination after feature reduction - all network sizes at different perturbation levels. Each feature is measured across 100 runs. 96

39 Categorization of peripheral and non-peripheral FFLs into canonical and embedded FFLs. An FFL is peripheral if the out degree of gene node (sink) is zero and is non-peripheral otherwise. 97

40 Distribution of peripheral and non-peripheral FFLs into canonical and embedded FFLs in E. coli transcriptional regulatory network. 98

41 Packet receipt rates (PRTs) for sampled transcriptional subnetworks of the bacterium *Escherichia coli* and *Saccharomyces cerevisiae* (labeled 'Yeast'). 104

42 Distribution of scaled feature importance values - 100 network size at different perturbation levels. Each feature is measured across 100 runs. . 105

43 Distribution of scaled feature importance values - 200 network size at different perturbation levels. Each feature is measured across 100 runs. . 106

44 Distribution of scaled feature importance values - 300 network size at different perturbation levels. Each feature is measured across 100 runs. . 107

45 Distribution of scaled feature importance values - 400 network size at different perturbation levels. Each feature is measured across 100 runs. . 108

46 Distribution of scaled feature importance values - 500 network size at different perturbation levels. Each feature is measured across 100 runs. . 109

47 Distribution of coefficient of determination before feature reduction - all network sizes at different perturbation levels. Each feature is measured across 100 runs. 110

Abstract

DISSERTATION

IN-SILICO MODELS FOR CAPTURING THE STATIC AND DYNAMIC CHARACTERISTICS OF ROBUSTNESS WITHIN COMPLEX NETWORKS

By Bhanu Kamapantula

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

Virginia Commonwealth University, 2015.

Director: Dissertation Dr. Preetam Ghosh,
Associate Professor, Department of Computer Science

Understanding the role of structural patterns within complex networks is essential to establish the governing principles of such networks. Social networks, biological networks, technological networks etc. can be considered as complex networks where information processing and transport plays a central role. Complexity in these networks can be due to abstraction, scale, functionality and structure. Depending on the abstraction each of these can be categorized further.

Gene regulatory networks are one such category of biological networks. Gene regulatory networks (GRNs) are assumed to be robust under internal and external perturbations. Network motifs such as feed-forward loop motif and bifan motif are believed to play a central role functionally in retaining GRN behavior under lossy conditions. While the role of static characteristics like average shortest path, density, degree centrality among other topological features is well documented by the research community, the structural role of motifs and their dynamic characteristics are not

well understood. Wireless sensor networks in the last decade were intensively studied using network simulators. Can we use in-silico experiments to understand biological network topologies better? Does the structure of these motifs have any role to play in ensuring robust information transport in such networks? How do their static and dynamic roles differ?

To understand these questions, we use in-silico network models to capture the dynamic characteristics of complex network topologies. Developing these models involve network mapping, sink selection strategies and identifying metrics to capture robust system behavior. Further, I studied the dynamic aspect of network characteristics using variation in network information flow under perturbations defined by lossy conditions and channel capacity. We use machine learning techniques to identify significant features that contribute to robust network performance. Our work demonstrates that although the structural role of feed-forward loop motif in signal transduction within GRNs is minimal, these motifs stand out under heavy perturbations.

CHAPTER 1

INTRODUCTION

1.1 Wireless Sensor Networks

Wireless Sensor Networks (WSNs) gather information from the deployed environment, which is processed and communicated to nearby nodes using a minimum of hardware: transmitters, receivers, a controller, and low-storage memory units. Large-scale WSNs are useful in military applications to monitor enemy targets, in disaster management to deliver critical environmental information, and in agricultural climate-monitoring applications. Despite these capabilities, they do not operate completely free of problems. Significant issues include transmission inconsistencies, channel noise, frequent hardware maintenance, reprogramming difficulties, and node failures. These issues increase the financial and energetic costs associated with more widespread implementation of such networks; reducing these costs requires breakthroughs in automated maintenance and repair, more efficient energy storage and use, and a focus on reducing error and mitigating sensor and packet damage. There are several parallels between genetic and sensor networks that motivate our discussions. Through a process termed transcription, genes process stimuli in the form of varying transcription factor levelsproteins responsible for activating/deactivating genesby producing mRNA molecules directly from the nucleotide sequence of the given gene locus. These transcription products may serve as the activating factors for other genes; so, genes "communicate" with one another by processing incoming signals (varying transcription factor levels) into output signals (the mRNA) used as input for the activation/deactivation of other genes. The network mapping com-

munication between genes in living tissues is termed the gene regulatory network. Similarly, wireless sensor networks (herein WSNs) are nets of communicating sensor motes, whereby hardware is responsible for processing incoming signals (the packets) into out-going messages (packet forwarding). Since living cells are able to adapt to disruptions to genetic "signals" due, in part, to the evolved network topology, we hypothesize that a deployed sensor network architecture based on GRN topologies will adopt similarly "robust" signal-transmission properties.

Studies have shown that the current *Homo Sapiens* have an estimated 250,000 years of evolution. However, the exact functioning of a Human body eludes us till date. Human body is an intricate system of complex mechanisms that continues to interest scientists including biologists, computational and medical researchers. To address this, an ambitious project called *The Human Genome Project* (HGP) was announced by the US Department of Energy in 1985. Its mission was to identify all the genes in the human genome. HGP was completed in the year 2003 [9]. However, the genome of every human being is unique and the data is still being refined to date [38]. Intensive research in structural genomics and their functional significance followed the completion of HGP creating the new field of Systems Biology, wherein the goal is to study the behavior and dynamics of complex biological systems.

Human body is made up of trillions of cells [47]. Each cell is comprised of genes in which information is encoded. The function of a cell varies depending on the level of gene expression that is regulated by a set of transcription factors. Such interacting genes and transcription factors can be represented as a Gene Regulatory Network (GRN). GRNs have been extensively explored by researchers as it is believed that they hold the key to unravel the mystery behind the working of a human body. Although a major portion of gene-gene interactions is still unknown for higher order organisms, the scientific community has recently focused on simulating the dynamics

of GRNs from lower order organisms. In such simulations, it is essential to consider the topological characteristics of GRNs that contribute to their robustness in information transport.

The GRN topologies considered in different parts of the work are that of *Escherichia coli* and the baker's yeast *Sachharomyces Cerevisiae*. A brief introduction to both the organisms of interest is presented in the following section.

1.2 Why *E. coli* and *Yeast*?

The bacterium *Escherichia coli* and the baker's yeast *Sachharomyces cerevisiae* have been widely studied by the biological research community. The genomes of both the organisms are mapped completely. Hereafter, *Escherichia coli* is referred as *E. coli* and the baker's yeast *Sachharomyces cerevisiae* as *Yeast*. Both these organisms are considered to be model organisms.

Following their genomic study, extensive studies were carried out by researchers to understand the transcriptional regulatory interactions between transcription factors and genes. These regulatory interactions can be represented as a network of nodes and edges where transcription factors and genes are the nodes and interactions among them are represented by the edges. The transcriptional regulatory network (TRN) of *E. coli* consists of 1565 nodes and 3758 unique edges. RegulonDB maintains the regulatory interactions within *E. coli* network [55]. Multiple studies have been carried out to understand the structural characteristics of this network [60, 31]. This network is scale-free in nature, and sparse. Scale-free property of a network is applicable when a large number of nodes have small degree (edges) and few nodes are enriched with large number of edges.

The following chapters are organized as follows. Chapter 2 briefly presents my research contributions. Chapter 3 introduces the concepts on robustness and the dif-

ferent contexts it is used before defining our metric for robustness. Chapter 4 defines several network characteristics derived from network motifs to capture robustness of a biological system. Chapter 5 presents the study on the role of vertex-shared motifs in network robustness. Chapter 6 demonstrates the structural role of feed-forward loop motif in signal transduction. Chapters 7 and 8 present the concluding comments and future work of this research.

CHAPTER 2

CONTRIBUTIONS

Biological networks offer a great opportunity to understand the governing principles of all the species and simultaneously, nature. These networks can be observed at different levels namely transcriptional regulatory networks, protein interaction networks, intracellular interaction networks among others. Transcriptional regulatory networks provide a good abstraction of a biological system by including the impact of transcriptional proteins on gene expression. Multiple reasons are suggested for this including the abundant presence of transcriptional motifs with functional and structural significance.

Wireless sensor networks (WSNs) are widely deployed now to sense environment in different areas including industrial monitoring, agriculture and civilian surveillance. WSNs, once touted to swarm the world are not without problems. They are plagued with signal disruptions and node failures.

The structural significance of substructures within biological networks are believed to have the advantage of evolutionary mechanisms which occur naturally. This dissertation motivates the need for designing engineered networks after reporting the studies carried out to identify robust network topologies and features responsible for the robustness. The outcome of this research will be identifying special features derived from transcriptional motifs which can then be used to design engineered networks. The main objectives of this research are outlined

1. Test the "robust" property of biological network topologies.

2. Map the gene regulatory network of interest to a wireless sensor network. Use

traditional network simulation platform to measure "robustness" of networks.

3. Studied and identified the structural patterns in the transcriptional regulatory networks of *E. coli* and *Yeast*.

4. Identify in-silico models to capture characteristics responsible for network robustness using motif-dependent features.

CHAPTER 3

ROBUSTNESS

With the rise of data on several fronts such as finance, social networks, biological networks etc in the recent decade, large number of studies were carried out to understand the consumption of this data. Studying this information using graph theoretic concepts emerged as an excellent approach. Here, essentially the underlying system is considered by a graph G where entities such as genes, users in social networks, officials in government, professionals in organizations are considered to be vertices (V) and the associations among them are represented as edges (E). The term robustness has been defined differently in different contexts. Biological robustness as defined in a breakthrough work by Hiroaki Kitano is "a property that allows a system to maintain its functions against internal and external perturbations" [32]. Robustness of *microRNAs* has also been studied in the biochemical networks, specifically their role in regulating certain hub nodes in interconnected modules under external and internal perturbations [50]. Nodes that have several connections to other nodes in a network are considered to be "hub" nodes. Modularity in biological networks has been shown to be critical to retain certain biological functionalities. Recent research has explored the principles behind the evolution of modules [14]. Several topological metrics were proposed and explored thoroughly to understand the structural patterns and redundancies in these networks. These metrics include centrality measures such as degree centrality, network centrality, eigenvector centrality, betweenness centrality and closeness centrality. Other metrics include average shortest path, network density, communicability, diameter. In a biological context, all these topological char-

acteristics are considered to be measures of how *robust* a system is. The next section introduces several metrics used in prior research to capture system robustness.

3.1 Metrics and definitions

Shortest paths between a pair of nodes captures the number of hops required to transmit information. Captured across the entire network, the average shortest path (ASP) measures the information transmission efficiency. If a given network has lower average shortest path, the network is *small* enough to send information using short paths. Shortest ASP is a defining feature of small world networks as introduced in [65]. For two vertices $V1$ and $V2$ in the network with V vertices, Equation 4.3 defines ASP as follows:

$$ASP = \frac{1}{|V|\left(|V|-1\right)} \sum_{V_1, V_2 \in V} \min\left\{d\left(V_1, V_2\right)\right\}. \tag{3.1}$$

Betweenness centrality (BC) defines the number of shortest paths passing through a specific node relative to all the shortest paths in the network. This captures the relative importance of a particular node compared to other nodes. This metric is often used to identify influential users in social network analysis. Given that $\sigma_{V_1 V_2}$ is the number of shortest paths between V_1 and V_2, Equation 3.2 defines BC of vertex V_i as follows:

$$BC_{V_i} = \sum_{V_1 \neq V_2} \frac{\sigma_{V_1 V_2}(V_i)}{\sigma_{V_1 V_2}} \tag{3.2}$$

Clustering coefficient (CC) is a measure of the ability to which nodes can in a network show a propensity to form clusters [48]. Communicability is a measure of capturing information transmission between two nodes and is used to identify communities within a network. For two nodes V_1, V_2, it is defined as the weighted

sum of all walks from V_1 to V_2 in which more weight is given to the shortest walks than the longer walks [11].

$$communicability(V_1, V_2) = \sum_{i=0}^{\infty} c_i (A^i)_{V_1 V_2} \tag{3.3}$$

where A^i is the $i-th$ entry for V_1 and V_2 in the corresponding adjacency matrix A of the network.

While these characteristics capture the topology of the respective networks, they do not capture dynamic aspects of biological networks. In addition to defining metrics such as network density/degree centrality, we also introduce several new metrics in Chapter 4 derived from feed-forward loop motif which is noted to play a crucial role in biological network robustness.

3.1.1 Robustness after attacks

Several studies have looked at the deterioration of complex networks by creating "attack" scenarios via deleting a set of nodes and edges and observing topological parameters such as the ones defined above. For instance, researchers in [6] identify the most influential edge set, delete them and observe the change in *natural connectivity*. The average eigenvalue of a network (Equation 3.4) is defined as *natural connectivity* [6].

$$natural\ connectivity = \ln(\frac{1}{V} \sum_{V_i \in V} e^{\lambda_{V_i}}) \tag{3.4}$$

Another study proposes using random walks in multi-layered networks and observes the variation in coverage ("average fraction of distinct vertices visited at least once in a time $< t$" [10]) under random failures. Research by [45] defines *robustness* as the number of remaining nodes after a cascading failure. As our interest is

in observing the dynamics of biological networks, we take a different approach by "simulating" a biological system.

3.2 Biological Networks

Various networks like social and information networks, genetic networks, economic networks can be considered as complex networks. The complexity in complex networks could be due to different reasons: scale, abstraction, functionality and structure. The scale of data in social networks brings great complexity to understand social dynamics of current society. Determining this will give insight on news and media consumption, face, contagion processes and virality. It has applications to the fields of artificial intelligence, machine learning, journalism, visual interface among others. Economic networks contain information regarding the money flow among global economies or banks; money flow among entities such as politicians, corporate interests; money and power flow among illegal entities. This is useful in understanding global debt, fraud detection and potentially illegal trade. Each of the problems mentioned above requires context, domain expertise, right data and computational algorithms to efficiently solve the problem. Our focus of interest are biological networks, specifically gene regulatory networks (GRNs).

3.3 Quantifying robustness in biological networks using NS-2

Our contribution lies at the realm of GRNs and in-silico experiments. We propose a framework to quantify biological robustness using NS-2, a network simulator. NS-2 has been primarily used to simulate different computer networks including Wireless Sensor Networks (WSNs). Information in this chapter is categorized as follows. Section 3.4 presents a discussion on the state of computational modelling of biological systems. Section 3.5 presents similarities and differences between GRNs and WSNs

thereby enabling a way to map a GRN to a WSN for simulation. Section 3.6 details the simulation setup including the parameters used and the assumptions. Section 3.6 also explains the network generation procedure and the sink selection strategies necessary for a network to be simulated. A case study is presented in Section 3.7 to identify the suitable model organism, between the bacterium *Escherichia coli* (abbreviated *E. coli*) and the baker's yeast *Saccharomyces cerevisiae*, for mapping purposes. Finally, future research directions are presented in Section 3.8.

3.4 Computational modelling

Ordinary differential equations (ODE) based computational models of biological systems, termed reaction rate equations or mass action kinetics, has received much attention [56]. Here, a homogeneous biological system is represented as a group of biochemical reactions and its dynamics are explored in the continuous-deterministic realm. However, ODE-based models are limited to study the underlying stochastic present in many biological processes such as gene expression and protein synthesis [15]. The limitations of ODE-based models for biological systems are detailed in [56].

[17] describes the advantages of using discrete event simulators for modeling biological systems. A fundamental challenge in computational systems biology [33] is the simplification of the biological system complexity without losing the ensemble dynamic behavior. In the system engineering view of complex processes [66], the key notion is to abstract the complexity of the system as a set of discrete time and space variables (random variables), which capture the behavior of the system in time. The entire system is a collection of functional blocks or modules, which are driven by a set of events, where an event defines a large number of micro level state transitions between a set of state variables accomplished within the event execution time. The underlying assumption driving this abstraction is the segregation of the com-

plete state space into such disjoint sets of independent events which can be executed simultaneously without any interaction. The application of this technique in large complex communication networks has demonstrated the accuracy of the approach for the first and higher order dynamics of the system within the limits of input data and state partitioning algorithms [67]. For example, discrete event based system modeling has been effectively applied for designing routers, the key components responsible for routing traffic through the Internet. Discrete event based simulation techniques have also been used in a wide variety of manufacturing processes and studying the system dynamics of complex industrial processes.

Researchers have also tried to adapt existing simulation platforms to model molecular communication. NanoNS is one such [18] simulation framework to model molecular communications. The framework is built over NS-2 software and uses a diffusive molecular communication channel. Researchers in [18] present an extensive review of communication models in nanoscale networks and out of three possible molecular communications, namely diffusive, motor-based and gap junction-based, their work is focused on diffusive-based molecular communication. As an extension of this work, researchers presented a case to build models for a variety of molecular communication channels, intra-body molecular nanonetworks and the network of such intra-body nanonetworks in [39]. This work comprehensively showcases the significance of modelling nanonetworks. Efforts are currently underway to simulate wireless nano sensor networks using NS-3 software (*next version of NS-2*) [52]. In this work, wireless nano sensor networks are modelled using electromagnetic communication instead of molecular communication as mentioned above. As it is evident by now, the challenges in achieving a simulation framework for communications in molecular networks are multifold [44]. Our core goal here is to identify ground rules for GRN-based robustness–the ability of a biological state to persist despite compo-

nent errors–by setting up a generic NS-2 simulation platform, rather than developing more detailed molecular communication channels.

Network simulator, NS-2 [22], is a discrete event simulator widely used for studying wireless networks. NS-2 has been used by researchers to model communication in wireless networks and embedded devices. This simulator continues to evolve with the active support of the research community. Taking a step forward, we have used NS-2 as an in-silico platform for quantifying the robustness of biological networks. Specifically, since the primary objective of a wireless sensor network is information transport to specific sink nodes, and because they operate under similar noisy and error prone conditions as biological networks, we define robustness of biological networks as the ability for each node in the network to deliver packets with minimal packet loss. Before envisioning a model for any time-varying functional biological system, it is important to illustrate the preliminary model for the biological system in NS-2. While exclusive simulators to model a molecular network are not present currently, existing simulators can be adjusted to model the desired network. It should be noted that this might not be the perfect approach, but the opportunity to explore the qualitative and quantitative dynamics of molecular networks is not lost. Scenarios are presented below whenever applicable to demonstrate the use of NS-2 to quantify biological robustness.

3.5 Mapping GRNs to WSNs

Transmission inconsistencies frequently plague WSNs where they suffer from signal disruptions due to sensor failure or from the absence of routing protocols that are sufficiently insensitive to local as well as global network conditions. In a WSN, nodes sense, process and communicate information with each other. Structurally, a GRN can be related to a WSN where every gene or transcription factor is a sensor. Signal

transmission within a GRN can be considered as packet transmission in a WSN. The fundamental assumptions in modelling a bio-inspired WSN are [26]:

1. GRN node structure is preserved in WSN.

2. Interactions among nodes in WSN are based on the existing connections in the GRN.

The physical signaling structure of sensors within the WSN must be adapted to reflect the communication between genes in the GRN. If gene $G1$ up-regulates $G2$, then the equivalent interaction in the WSN is that sensor $S1$ sends a packet to $S2$ according to specific probability distribution defined by gene-gene interactions. For homogeneous sensor nodes, each up-regulation edge in a GRN is replaced by a bi-directional edge; if we allow sensor $S1$ to send a packet to $S2$, then $S2$ should also be able to send a packet to $S1$. For heterogeneous sensor nodes, however, it is not necessary that both $S1$ and $S2$ possess the same transmission radii, giving a directed edge from $S1$ to $S2$ and not vice versa.

We recognize that WSNs conceptually operate under noisy and/or adverse conditions similar to the stochastic cellular environment encountered by GRNs. We hypothesize that if it is possible to exploit the simulation platform used for WSNs, namely NS-2, to assess the signal transmission robustness in GRNs, then any observed robust qualities can be explained by fundamental biological processes, such as transcription. The process where signals from nearby neighbors in the form of transcription factors stimulate/inhibit other genes by generating mRNA molecules is transcription. Thus, GRN nodes communicate with one another by sending signals (transcription factors), which are in return processed into output signals (mRNAs). This process is similar to WSNs where sensors receive packets from its neighbors with packet forwarding instructions to other destination nodes. As a result, any node in a

network (GRN or WSN) can affect the decision of other nodes and hence the overall network performance.

Here, we considered the transcriptional regulatory network (TRN) of the bacterium *E. coli* to generate the sample GRN graphs. Such TRNs bear the actual topology of the GRNs with any gene-gene and gene to transcription factor edges deleted. Thus, in such TRNs, a single transcription factor can regulate other transcription factors and genes, while genes do not directly regulate other nodes. Note that our earlier work on WSNs derived from GRN topologies actually considered the TRNs from *E. coli* which were shown to achieve high packet transmission efficiency [26]; hence such TRNs exhibit the desired biological robustness measures that we seek to model here. The transcription factor molecules having half-lives $T_{1/2} = \ln 2/k$ [3], where k represents the decay rate constant, are subject to degradation if held at the transcriptional regulation queue. Similarly in the case of WSNs, packets are forwarded from source nodes to sink nodes using multiple hops and can be dropped at intermediate nodes if they exceed the queue length. Hence, genes can be considered as sink nodes and transcription factors as the source nodes. On that account, we describe our measure of robustness in WSNs that adopt the GRN topologies as the ability for each node in the network to deliver information to their local sinks with minimal packet loss.

3.6 NS-2 simulation setup

Consider a biological network topology derived from a well studied organism, *E. coli*. Sub-networks that are extracted from *E. coli* comprise of interactions among genes. Let us call this extracted network a Gene Regulatory Network (GRN). Such GRNs comprise two classes of nodes: transcription factors and genes. A transcription factor either up-regulates or down-regulates one (or more) gene. The packet

transmission rates are assumed to be identical in NS-2, for all the non-sink nodes; however, in a real biological setting, such rates are directly proportional to the rate constants associated with every edge in the network along with the concentration of the molecules associated with a node. This however creates a roadblock for existing biological network simulators as each of these rate constants need to be experimentally validated which is not currently feasible for the different sample networks generated in this work. The simulation also assumes all packets transmitted to be identical in type and size which correspond to similar signaling molecules affecting the different nodes in the GRN in the context of biological robustness.

Queue limit in NS-2 is useful to limit the number of packets that can be queued at a node. Queue limit in the corresponding GRN represents the half-life of each signal sent from one node to another node. Although this is another approximation in the simulation set-up, it is impossible to characterize all such signaling molecules accurately in the different extracted GRNs. In summary, our proposed NS-2 set-up makes broad assumptions for the pertinent details of biological network signaling but we feel that this is indeed necessary for studying the qualitative dynamics of many sample GRNs wherein such details are not known at length.

Traditionally, robustness of biological networks has been measured by its static graph theoretic characteristics such as network diameter, average shortest path [46], network efficiency [36] amongst others. A network with negligible change in its diameter is considered to be robust when it loses node(s) after an attack. Similarly, negligible change in average shortest path and network efficiency under network perturbations related to temporal fluctuations in the node and/or link availability is attributed to robust networks. Packet receival rate is the ratio of the number of packets received in the network to the number of packets sent. Higher the packet receival rate of a GRN, higher its robustness. Randomly generated WSNs and GRN-

derived networks are compared with respect to the packet receival rate. This section discusses the methods used for random network generation to be used as a wireless sensor network. In addition, approaches used to identify the sink node in the network are detailed. A new algorithm for biological network generation is presented in Section 3.6.2.1.

3.6.1 Network generation

A script written in the Python programming language [54] is used to generate networks modelled as WSNs. Here, two different nodes within the network are chosen at random, and a link is established between them with probability p [1]. Networks with 100, 150, 200, 250 and 300 nodes were generated for demonstration purposes as representing "medium" sized sensor networks. 25 networks of each size (100, 150, 200, 250 and 300 nodes) are considered to illustrate the sink node selection approach. Networks of a certain size are spread over an area with specific node transmission range. For example, 25 different networks of size 150 nodes are spread over $3{\cdot}6\times10^5\text{m}^2$ (with x=600m and y=600m) with a node transmission range of 85 meters. Node range for a network has been assigned based on the work by [19]. Similarly, networks of size 200 are spread over area of $4.9\times10^5\text{m}^2$ (with x=700m and y=700m) with a node transmission range of 90 meters and networks of size 250 are spread over $8.1\times10^5\text{m}^2$ (with x=900m and y=900m) with a node transmission range of 90 meters. Networks of size 300 are spread over an area of 10^6m^2 (with x=1000m and y=1000m) with a node transmission range of 110 meters. Few assumptions are made for simplicity. The directionality of the links between the nodes is ignored. Self-edges [2], edges with

[1] $p(K)$ is the probability to find a node of degree K in a network that follows the power law distribution $p(K) \sim K^{-\gamma}$.

[2] In a biological context, self-edges for a gene refers to auto-regulation of expression.

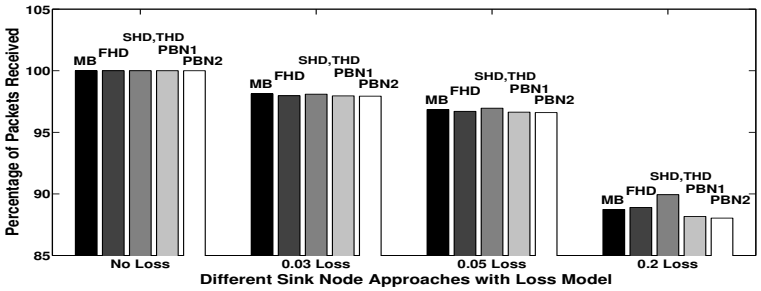


Fig. 1.: Sink node selection and respective packet receival rates for different loss models - GRN of 20 nodes [30]

same source and destination nodes, are removed from the network. Nodes in model organisms such as *E. coli* and S. cerevisiae self, up- or down-, regulate themselves. However, we ignore self-edges in this case of WSN simulation. In order to compare similar entities, only networks with same number of nodes and edges are considered for comparison. All 25 networks of the same size have exact number of edges. Each network generated using this approach is considered to be a Random Wireless Sensor Network (RWSN).

3.6.2 Sink selection strategy

Sink node selection strategy is critical for optimal GRN performance. In [28], we listed three sink selection strategies: (a) Highest Degree (HD), (b) Highest Coverage (HC) and (c) Motif-based (MB) and identified HD strategy as the best approach to provide higher robustness for NS-2 based simulation of GRNs. Nodes with highest degree are selected as a sink node in the HD strategy. Node involved in any three-node motif is selected as a sink node in the HC strategy. Figure 1 shows the comparison of sink selection strategies for a GRN-derived of twenty nodes [26]. In this figure, FHD stands for the node with First Highest Degree, SHD stands for the node with Second Highest Degree, THD stands for the node with Third Highest Degree and PBN stands

for node identified with Probabilistic Boolean Network. A PBN is a formalism where set of functions define the expression value of genes in the network. The node with the highest expression is selected as sink node. For detailed information on sink selection strategies including PBN, refer to [26]. Intuitively, using FHD node as a sink makes sense since the node is regulated, in a biological context, by several other regulators and are critical for important biological functionalities. Such nodes also act as hubs in a network.

Three-node motifs have been earlier identified as the building blocks of robust GRNs [43] from a purely topological perspective, and the feed-forward loops, wherein two genes regulate each other and they both regulate a third, were reported to have the most significant impact on GRN robustness. Hence, we also considered nodes involved most in a feed-forward loop (FFL) motif as a sink node in the MB strategy. We considered FFL motifs as they have been identified to play an important role in establishing robustness [34] apart from ensuring important biological functions such as generating signal pulses, and speeding up or delaying response times in target genes [40].

In [26], we compared several GRN-derived networks with randomly generated networks (network sizes 100, 150, 200, 250 and 300) and showed that GRN-derived networks improve the transmission reliability in our NS-2 based simulation setting. The procedure for generating random networks is described in Section 3.6.1. Figures 2 and 3 present a comparison for best, mean and worst performing RWSNs and GRN of network sizes 100 and 300 respectively. For this experiment, a total of 25 RWSNs are considered and three cases are presented. Comparisons are also made for large-scale predicted GRNs (network size 1500, 1750, 2000, 2250 and 2500). The performance of GRN vs RWSNs in large scale networks (network size 1500 and 2500) is presented in Figures 4 and 5. The graphs for network sizes 1750, 2000 and 2250 are not reported

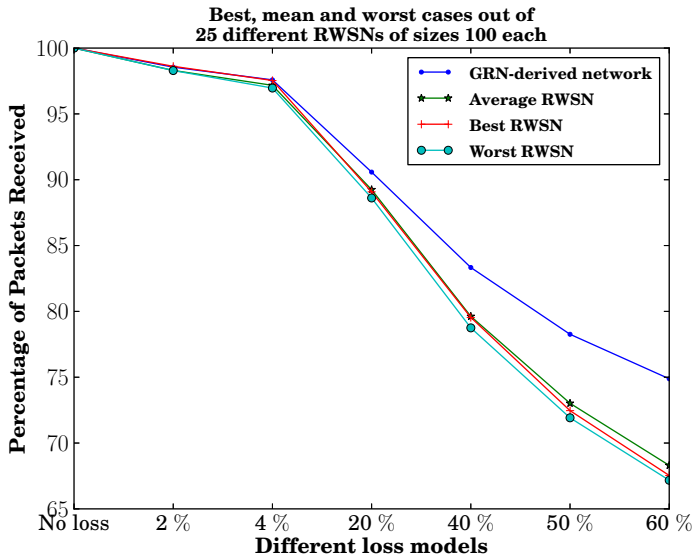


Fig. 2.: Comparison of best, mean and worst (out of 25 networks) performing RWSNs to GRN - Network size 100 [26]

since they follow similar trend as networks of size 1500 and 2500. This might be possible due to the presence of higher number of FFLs in GRN-derived networks as compared to randomly generated networks. The abundance of FFL motifs in random networks and networks derived from new algorithm [3] is presented in Table 1. The counts reported in the table are averaged, and approximated to nearest decimal, across ten different networks of a particular type.

3.6.2.1 New network generation algorithm

Here we discuss the network generation from our work in [41]. For brevity, the Scale-free Directed Network Generator is referred to as SDNG). The algorithm can be utilized to expand existing networks as well as generating directed networks emulating the different distributions of *E. coli*, namely in-degree, out-degree, cumulative degree

[3]Algorithm proposed by [41] is explained in Section 3.6.2.1.

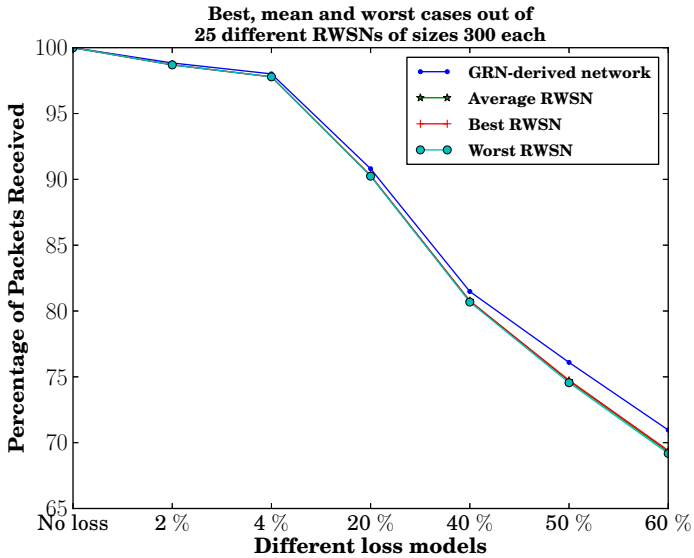


Fig. 3.: Comparison of best, mean and worst (out of 25 networks) performing RWSNs to GRN - Network size 300 [26]

Table 1.: Feedforward Loop Motif Count in RWSNs and new algorithm networks [26]

Network Size	**FFL Count in RWSN**	**FFL Count in New algorithm networks**
1500	3972	8429
2000	4125	8524
2500	6742	8591

and the participation of genes in feed-forward loops. The algorithm is similar to the Barabási-Albert (BA) model which uses the preferential attachment mechanism [2] for growing scale-free (SF) networks. Networks are grown resembling the phenomena known as the 'rich get richer and the poor get poorer', however the BA model was originally employed for undirected networks. The duplication-divergence (DD) model suggested by [64] considers the growth of directed biological networks. The suggested model which was later extended in [8] was predicated by the fact that proteins/genes evolve through copying themselves followed by their subsequent infrequent mutation. In addition to using the cumulative distribution as the sole measure for resembling the original networks, few of the DD grown networks retained a power-law distribution.

To illustrate the dynamics of SDNG, we consider denoting candidate nodes for preferential attachments in an existing network of size n with subscript i, wherein K_i and R_i label the out- and in-degrees respectively. The probability for a candidate node to be connected to a node foreign to the existing network with an edge directed from the candidate node to the foreign node is given by $A(K_i, R_i)$. The probability that a link is drawn from the foreign node to the candidate node is given by $B(K_i, R_i)$. Each probability is normalized against all nodes of the existing network to form attachment kernels [35], and their formulas are listed in Table 2.

For this particular work, we considered the power-law attachment kernel for calculating the edge probabilities. Starting with a fully connected eight node network, a candidate node is picked at random with equi-probability. Next, a random number d is selected with equi-probablity from the interval $d \in (0, 1)$. An edge is drawn from the candidate node to the foreign node if $d \leq A(K_i, R_i)$. This process is then repeat for an edge drawn out of the foreign node to the candidate node, provided the probability satisfies $d \leq B(K_i, R_i)$. The above steps are then reiterated $m_i - 1$ times, wherein m_i is an another number selected at random from an exponential probability

Table 2.: Attachment kernels

Functional Type	**Attachment Kernels**	
	$A(K_i, R_i)$	$B(K_i, R_i)$
Linear	$\frac{K_i}{\sum_{i=1}^{n} K_i}$	$\frac{R_i}{\sum_{i=1}^{n} R_i}$
Power-law	$\frac{K_i^{0.8}}{\sum_{i=1}^{n} K_i^{0.8}}$	$\frac{R_i^{0.8}}{\sum_{i=1}^{n} R_i^{0.8}}$
Sigmoid	$\frac{K_i}{\sum_{i=1}^{n}(K_i+R_i)}$	$\frac{R_i}{\sum_{i=1}^{n}(K_i+R_i)}$

distribution $\rho(m_i) = (f^{\frac{1}{1-m_0}} - 1)f^{-m_i/(1-m_0)}$. The decay of this distribution resembles the degree distribution of *E. coli*. Here, we considered values of $f = \frac{1}{4}$ and $m_0 = 2$.

3.6.3 SVM Validation

While the network evaluations presented in Figures 2 and 3 establish the significance of GRN-derived networks, only one sink operates in those networks which is not the case in functional GRNs. To address this, we used multiple sink nodes to model GRN communication. An Support Vector Machine (SVM) model, built using LibSVM [7], is then used to investigate the relative efficiency of packet receival rates based on topological metrics such as network density, genes coverage, transcription factor network density, motif abundance and genes percentage, defined below.

For this, GRNs of varying sizes, $100<n<500$ were used, where n is the number of nodes in the GRN. Transmission is considered from source nodes (similar to transcription factors) to sink nodes (similar to gene nodes). 410 out of the 490 networks are used to train the learning model and remaining networks are used to test

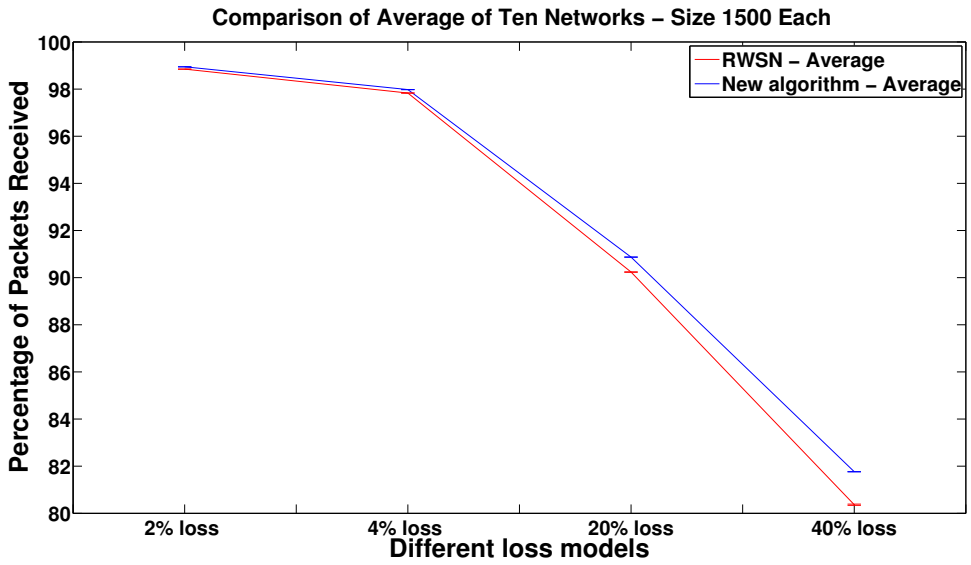


Fig. 4.: Comparison of average of ten networks between New Algorithm and RWSN - Network size 1500

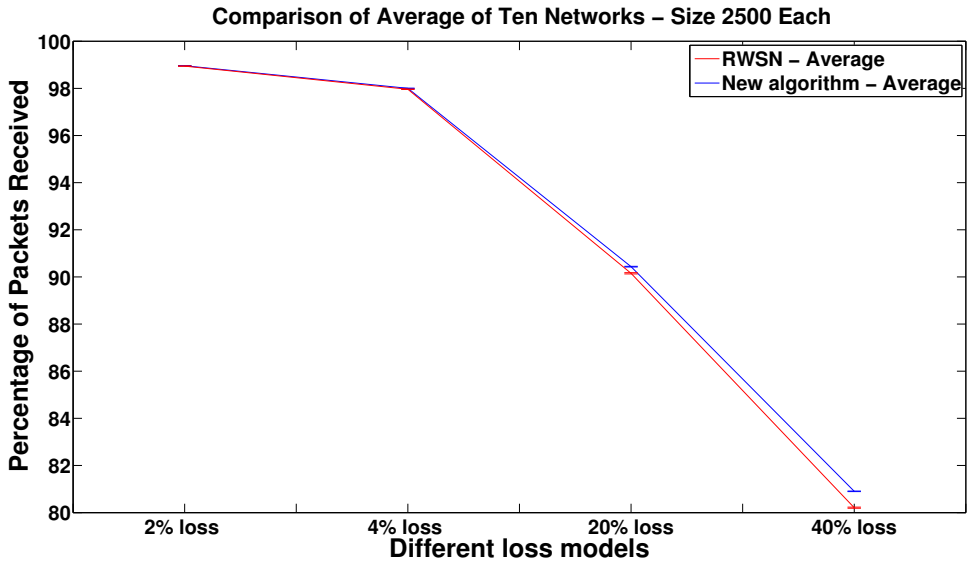


Fig. 5.: Comparison of average of ten networks between New Algorithm and RWSN - Network size 2500

the model. The directionality of the links between the nodes is considered. The topological metrics used in the learning model are briefly described below.

3.6.3.1 Network density (ND)

ND is a ratio of the number of edges present in the network to the total number of edges possible in the network.

3.6.3.2 Genes coverage (GC)

GC is the summation of the ratios of in-degree of each sink node to the ratio of source nodes having a path to that particular sink node.

3.6.3.3 Transcription factor network density (TND)

TND is the ratio of the number of edges that transcription factor nodes participate to the total number of edges in the network.

3.6.3.4 Motif abundance

Motif abundance is the ratio of abundances of FFL (R^{FFL}) and bifan (R^{BF}) motifs that relate to the number of nodes.

3.6.3.5 Genes percentage (GP)

GP is the ratio of number of gene nodes to the total number of nodes in the network.

3.6.4 Contributions of topological metrics to GRN robustness

These topological metrics are then used to construct the SVM learning model. Cross validation is used in the training stage; test data is then used to predict the

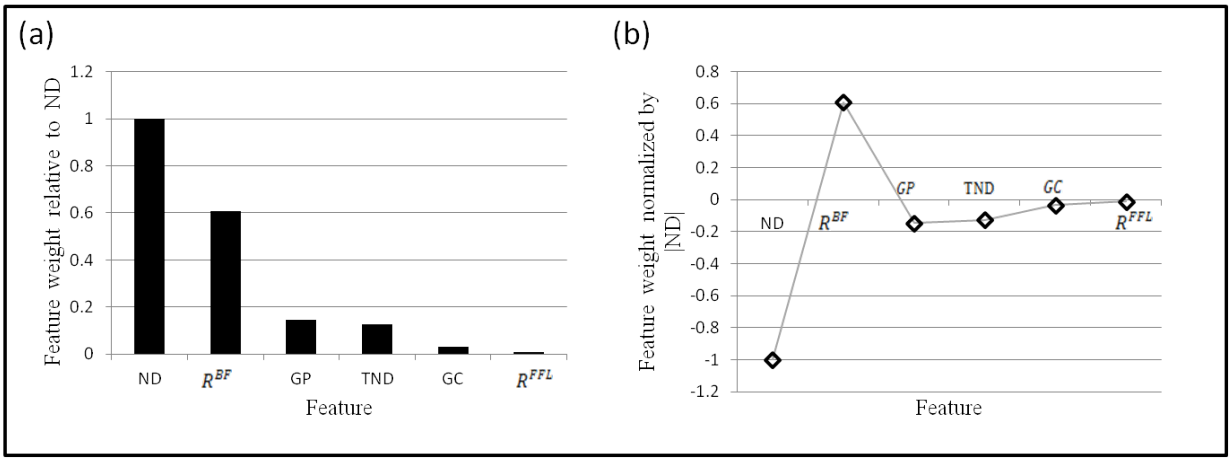


Fig. 6.: (a) Relative importance of the feature weights, (b) Relative importance of feature directions

robustness of the networks. The relative importance of the features used in the model in the decreasing order is as follows: ND, R^{BF}, GP, TND, GC and R^{FFL}. Figure 6(a) shows the weight w_i of features divided by the maximum weight (w_{ND}): $|wi/w_{ND}|$. Figure 6(b) shows same ratio but the directions of the weights are considered. It should be noted that a GRN is more communicative when it is sparse implying low ND and high R^{BF} as shown in Figure 6(b).

3.7 Case study: Comparison of derived networks from E. coli and Yeast

We have demonstrated the performance of NS-2 as a platform to quantify robustness in biological networks. In order to exploit the principles of a biological network, it is crucial to evaluate the model organisms. For this purpose, we compare networks derived from two well studied model organisms, *E. coli* and S. cerevisiae, of sizes consisting 100, 200, 300, 400 and 500 nodes using GeneNetWeaver software [57]. One hundred networks of each size are generated and NS-2 simulations are performed on each of these networks. As comparing the average performance of all networks may

not distinguish the performance of the derived networks properly, we compared the best perfoming, average performing and least performing networks. The directionality of the links between the nodes is ignored. The simulation parameters are as follows:

1. Bandwidth = 1Mb
2. Delay = 1.0ms
3. Queue limit = 5
4. Packet size = 900 bytes

Figure 7 shows the best performing derived networks from *E. coli* and S. cerevisiae for network sizes: 100, 200, 300, 400, and 500 (nodes) w.r.t. 20%, 35% and 50% loss. While the performance of S. cerevisiae derived networks is consistently higher for 500 node network under 20% and 35% and 50% loss, *E. coli* derived networks perform better, in almost all cases except for 200 network size at 20% loss, for networks of size 100, 200, 300 and 400.

Figure 8 shows the mean performing derived networks from *E. coli* and S. cerevisiae for network sizes: 100, 200, 300, 400, and 500 (nodes) w.r.t 20%, 35% and 50% loss. It can be clearly observed from the figure that the performance of *E. coli* derived network is better at 20% and 35% loss and S. cerevisiae derived network performs better for higher loss percentage (50%). The difference in performance is ~ 0.51 at 20% loss (*E, coli*), ~ 0.38 at 35% loss (*E. coli*), ~ 0.359 for 50% loss (*Yeast*). It appears that S. cerevisiae derived network performs better than *E. coli* derived network at higher loss percentage.

Similarly, Figure 9 shows the worst performing *E. coli* and S. cerevisiae derived networks for network sizes: 100, 200, 300, 400, and 500 (nodes) w.r.t 20%, 35% and

50% loss. It can also be noticed here that S. cerevisiae derived network performs better than *E. coli* derived network only for higher network size (500 nodes) and the latter performs better than the former for other network sizes (100, 200, 300 and 400 nodes).

Figure 10 shows the comparison of packet receival rates for networks of size 100 (nodes). The difference in the packet receival rates of the best performing *E. coli* and S. cerevisiae derived networks suggests that *E. coli*-derived network performs better than yeast-derived network. Figure 11 shows the comparison of packet receival rates for networks of size 500 (nodes).

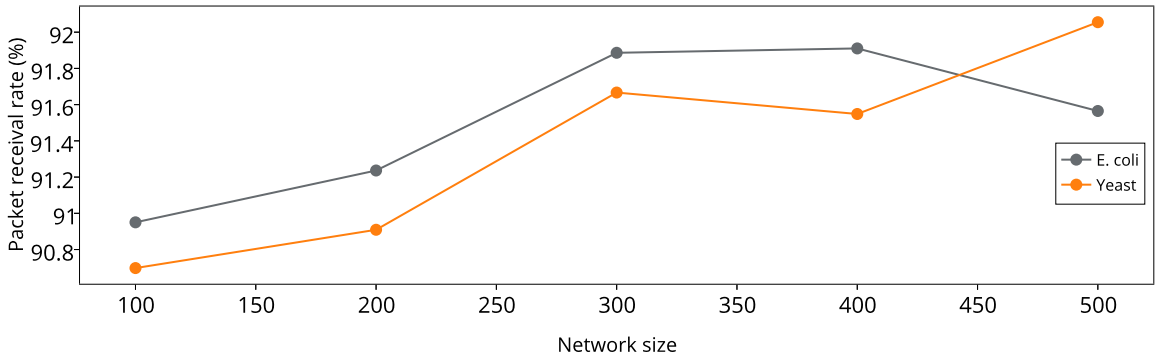
To arrive at any decisive conclusion on a better model organism for WSN mapping, extensive simulations need to be performed to check if this trend holds for higher network sizes (1000 or 1500 or 2000 node network etc.). Since S. cerevisiae performs marginally better at a high loss rate, sparse WSNs in real-world applications − where communication is essential even at high loss, for instance, during rescue operations after natural disasters − can be modelled using the structural principles of yeast-derived GRN.

Our simulation setup using NS-2 is generic and can be applied to any GRN (e.g: *E. coli*, S. cerevisiae), and thus provides a common platform to assess dynamic robustness of biological networks. This also allows to sample several extracted and predicted GRN topologies and measure their signal transmission dynamics thereby identifying specific topological and control properties in these networks that impact their robustness. Such a platform will hence allow one to compare the robustness of the GRN topologies of different organisms, design, validate, test and explore different GRN prediction algorithms besides also serving the greater complex networks community by applying such design rules of robust biological networks to create fault-tolerant and efficient engineered systems.

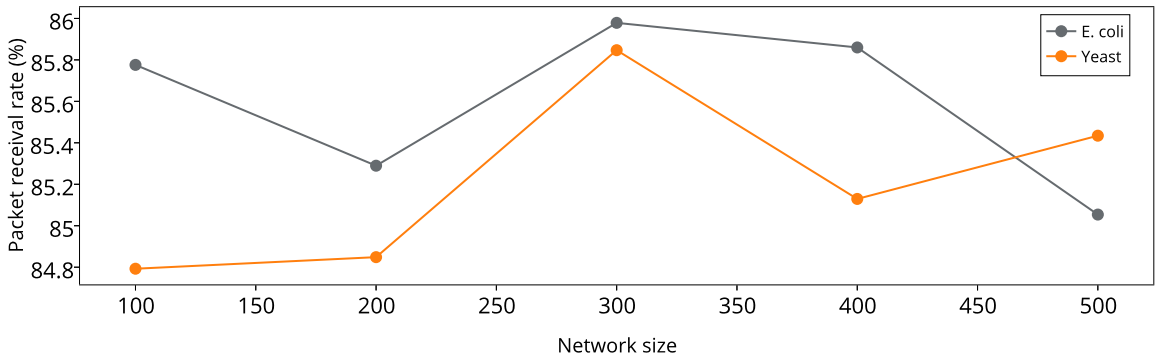
3.8 Challenges and Future directions

NS-2 is a discrete event simulator built for exploring wired networks and then extended to study wireless networks. It is not exclusively built for communication in molecular networks. Creating an environment for simulating molecular networks is extremely challenging. In a biological network, transmission of signals from one node (transcription factor) to another (gene/transcription factor) occurs at a rate that has not been determined yet. Active effort by researchers is focused on estimating such rate constants. Determining the rate constants is critical for modelling the dynamic behavior of a biological system. While our work is preliminary, it allows us to qualitatively and quantitatively simulate biological networks (specifically GRNs) without any knowledge of the underlying rate constants. This will help in establishing the reasons behind the inherent robustness of GRNs as well as motivate the design of efficient WSNs, wherein routing algorithms that intuitively embed biological structural properties in WSNs need to be developed. This can be realized using repeating structural patterns in biological networks termed as motifs.

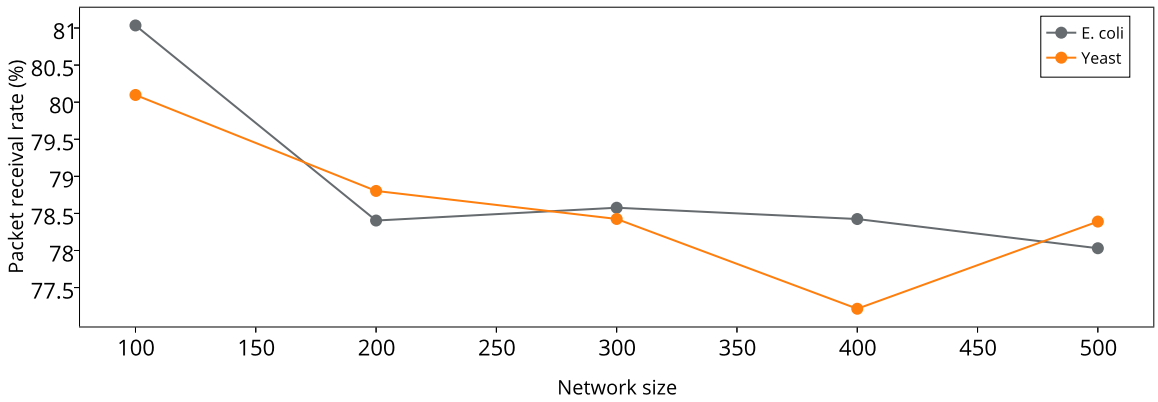
Work is currently underway to identify features derived from feed-forward loop motifs. Following this, A WSN can be categorized into several pockets of such patterns and routing can be introduced from different nodes to the sink to achieve higher packet transmission efficiency. Adaptive routing mechanisms can be imagined to improve WSN efficiency. Bandwidth limitations on edges and nodes in a regulatory network need to be studied before bandwidth based studies can be carried out in WSNs. Much needs to be realized in this field before a true bio-inspired WSN is modeled that adheres to structural and dynamic behavior of a biological system.



(a) 20% loss

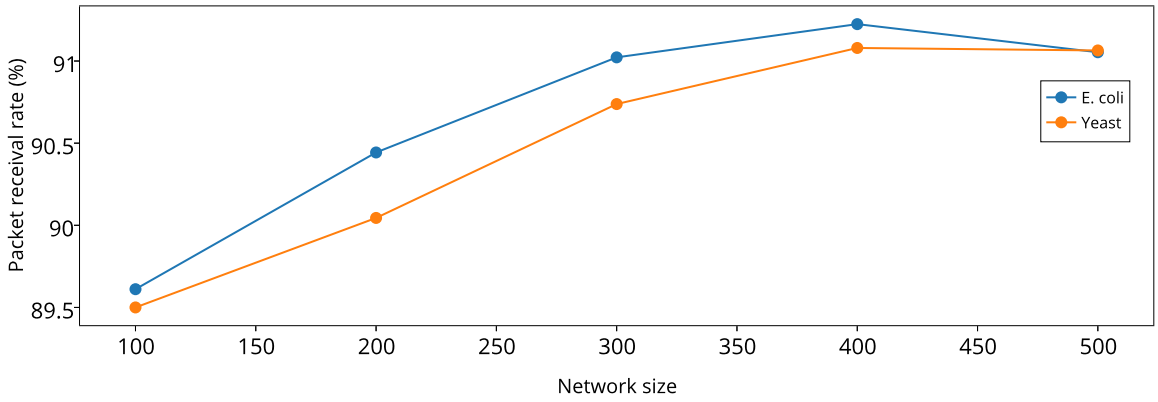


(b) 35% loss

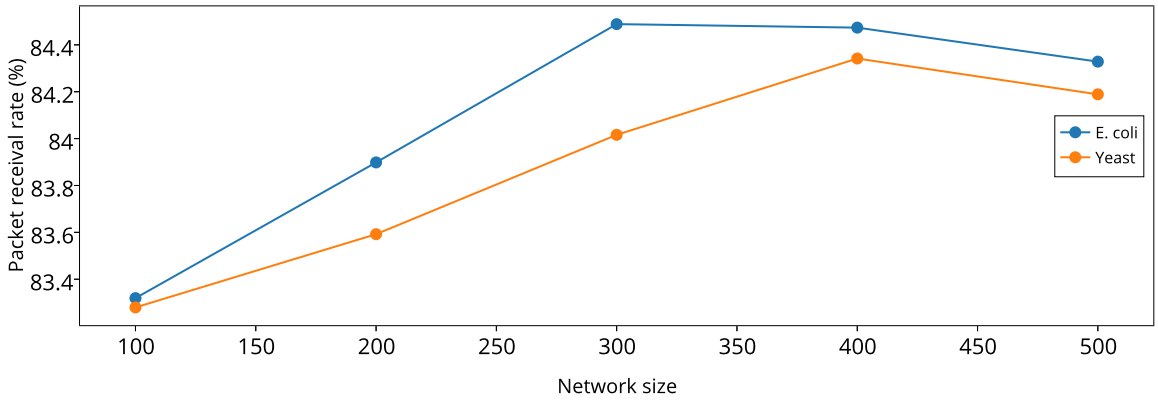


(c) 50% loss

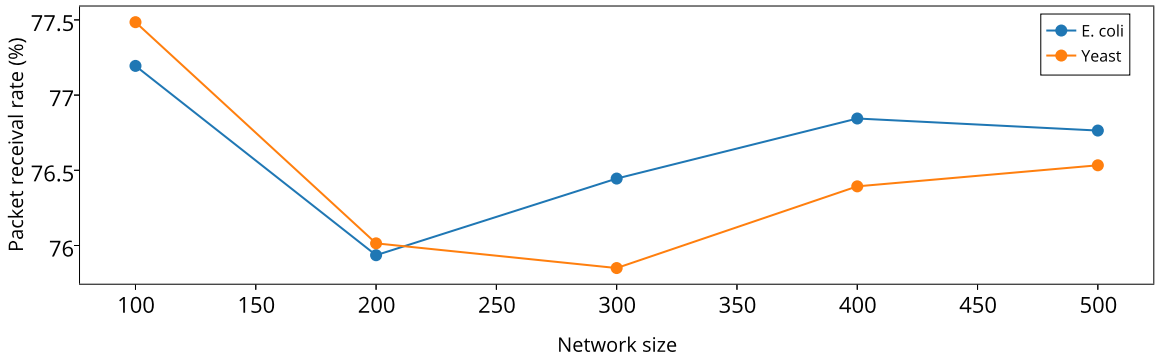
Fig. 7.: Comparison of best performing networks derived from E. coli and Yeast - 20%, 35% and 50% loss



(a) 20% loss

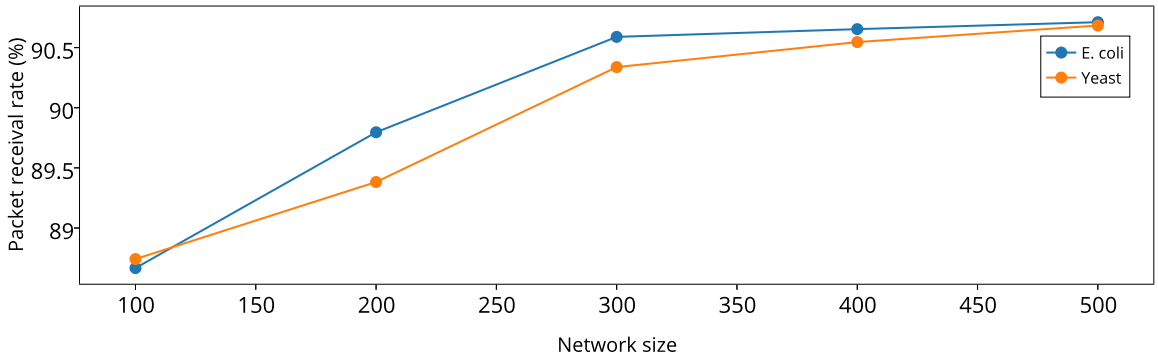


(b) 35% loss

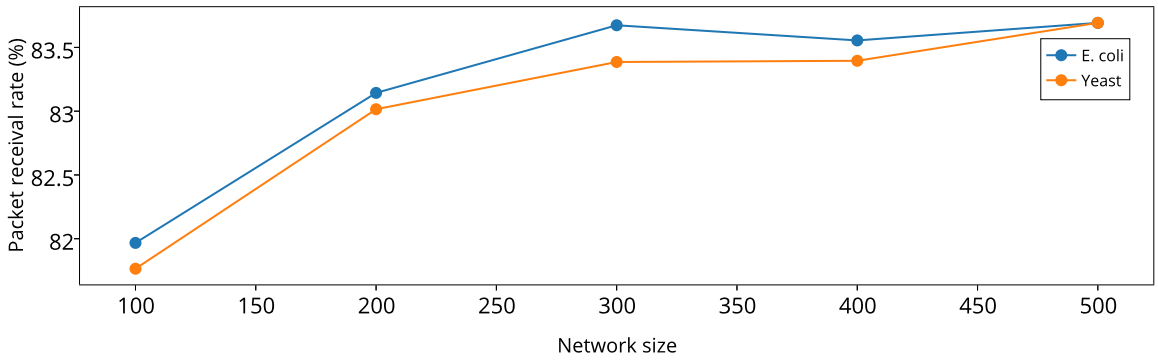


(c) 50% loss

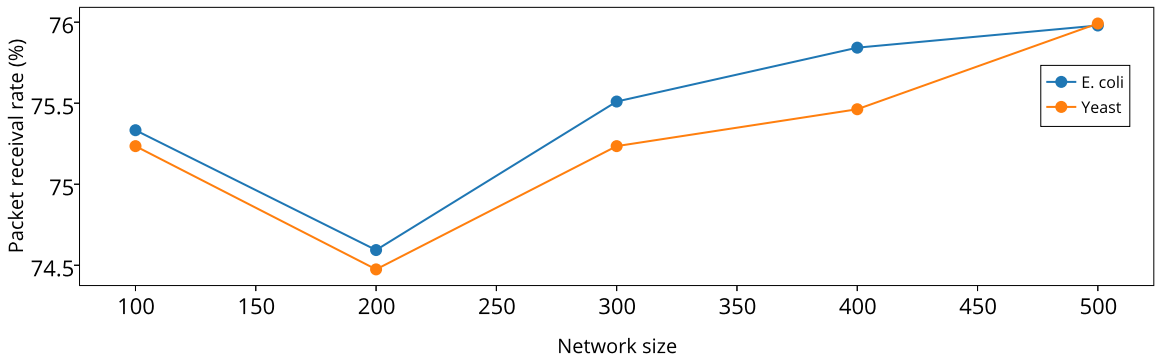
Fig. 8.: Comparison of mean performing networks derived from E. coli and Yeast - 20%, 35% and 50% loss



(a) 20% loss



(b) 35% loss



(c) 50% loss

Fig. 9.: Comparison of worst performing networks derived from E. coli and Yeast - 20%, 35% and 50% loss

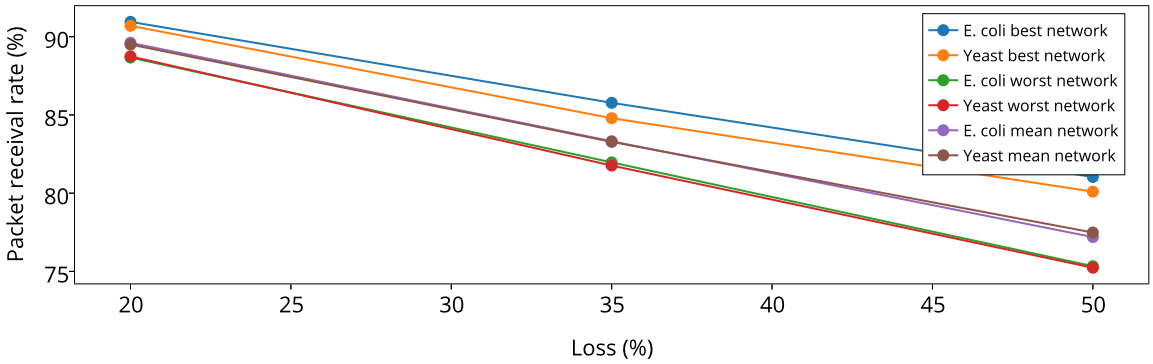


Fig. 10.: Comparison of 100 node networks derived from E.coli and Yeast respectively - 20%, 35%, 50% loss

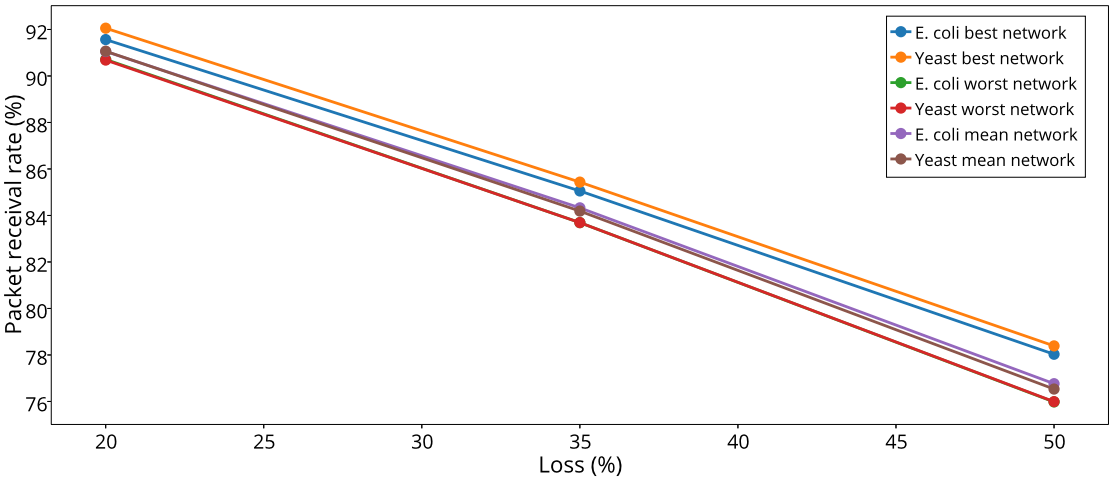


Fig. 11.: Comparison of 500 node networks derived from E.coli and Yeast respectively - 20%, 35%, 50% loss

CHAPTER 4

STRUCTURAL REDUNDANCY OF TRANSCRIPTIONAL MOTIFS

4.1 Introduction

Many functional aspects of transcriptional networks appear to be preserved despite the presence of noise or other disruptions. For example, some bacteria have been shown to survive despite extensive 'rewiring' of their transcriptional network topologies [23]. In some cases, such a robustness to function can be attributed to the network structure alone, owing to its power-law degree distribution [1]. In other cases, the abundance of highly repetitive subnetworks, termed transcriptional motifs [60], have been correlated with an ability of the system to persist in a dynamically stable state [53]. One interesting example of a transcriptional motif is the feed-forward loop– a small, three-node subnetwork wherein the top-level protein regulates the expression of a gene via two paths, which appears to be more abundant in some transcriptional networks than found in randomized versions [60]. Indeed, feed-forward loops have received much attention, due in part to their information-processing ability. For example, they have been reported to speed-up or slow-down response times without any feedback loop [40].

This ability to remain useful despite experiencing significant disruptions to communication seems to be a generic property of biology [34], and finding general properties or 'laws' that can be used to engineer this feature into man-made systems remains a 'holy grail' of systems architecture and control theory [37]. We make headway toward this goal by using machine learning techniques to interrogate the relationship between topological and dynamical properties of transcriptional networks, but viewed

from the angle of the application; in this case, a scalable wireless networking system. Here, nodes with communication capacity may continually enter or leave the system, which has parallels in molecular biology: proteins and other signaling biomolecules are continually made and destroyed, leading to uncertainty in the channel capacity of a signaling pathway. Our approach to this problem is to combining discrete event simulation and support vector machine learning techniques to identify important system features that contribute to the information flow across such networks. Discrete event simulation can capture dynamic behavior of the system by modeling information transmission as a set of independent events under custom perturbations using channel noise and congestion-based information loss; machine learning techniques can be used to identify underlying patterns in the data.

The NS-2 framework simulates information flow across wireless man-made systems in terms of packet transport, and we employ it here to quantify a type of dynamical network robustness by measuring the packet receipt rates at various destination nodes in the model networks. Packet receipt rate is determined as the ratio of number of packets successfully received at sink/destination nodes to the number of packets sent by the source node(s). While biological systems do not strictly communicate using information packets, they do employ signal transduction pathways that can be thought of a series of activation steps or 'checks,' which succeed upon passing a concentration threshold. This analogy can be taken further, given that biology is often redundant, in the sense that many pathways may be activated to achieve a single goal, reminiscent of flooding. We have described such similarities in detail before [16, 28, 27].

The results reported here build upon our previous work to explore properties crucial for robustness in transcriptional networks to design specialized wireless sensor network topologies [16, 28, 27], and quantifying performance of such networks using

the NS-2 simulation framework [30].

4.2 Methods

4.2.1 Model transcriptional networks

The GeneNetWeaver software package [57] is used here to extract subnetworks from transcriptional network datasets for the bacterium *Escherichia coli* and the common baker's yeast *Saccharomyces cerevisiae*. One hundred networks of five different network sizes $n = 100$, 200, 300, 400, and 500, as represented by the number of nodes n. For simplicity, we will refer to networks derived from *S. cerevisiae* as 'Yeast' networks, whereas the bacterial networks will be referred to as *E. coli* networks. For our purposes, we map the transcription factors as nodes, and transcriptional network edges represent are understood to denote interactions between participating nodes; thus, we ignored the regulatory interaction of each link. As a result, we may apply the concepts of graph theory [4] to the resulting networks.

4.2.2 Simulation setup

Network simulator (NS-2) software [42] is used here to simulate packet transmissions in the mapped network. Nodes corresponding to genes that code for transcription factors in the genetic network are taken as the source nodes, whereas nodes corresponding to nonregulating genes are considered to be the sink nodes. While source nodes can send and forward packets, sink nodes may only receive packets without forwarding them onto others.

A queue limit of five packets is arbitrarily set for each participating node in the network simulation; we adopt a flooding type protocol, wherein each node may send ten packets each to its outgoing edges. Thus, non-sink nodes with outgoing edges

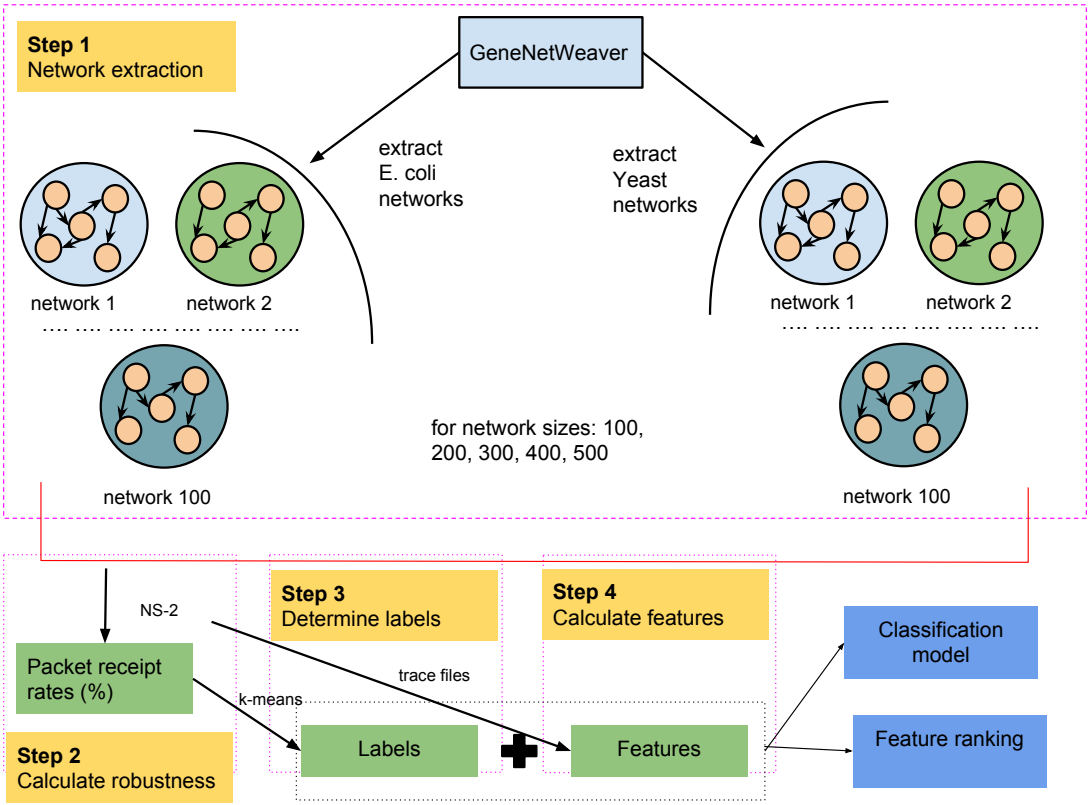


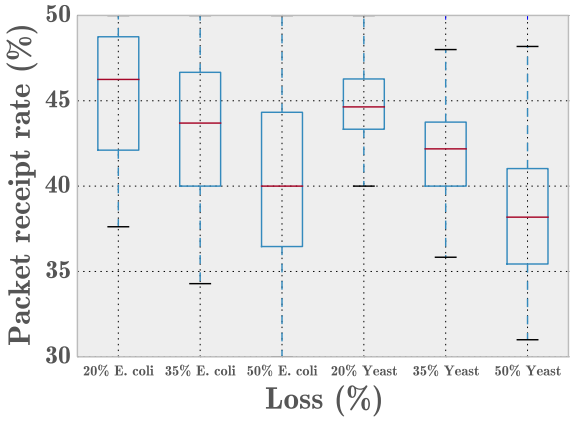
Fig. 12.: Procedure followed to identify significant features in *E. coli* and *Yeast* sub-networks

forward packets until the simulation ends.

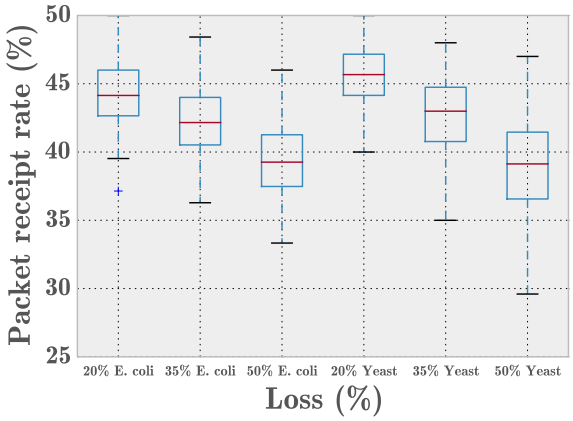
To account for noise, three different loss scenarios are considered, in which up to 20%, 35% and 50% of packets can be "lost" in transit. This affects the packet receipt rate, which is determined to be the ratio of number of packets received at all sinks to the number of packets transmitted by source nodes, which, for convenience, we represent as a percentage of the total sent packets: (packet receipt rate)$\times$100. This dynamical system is perturbed by fluctuating the loss level. Since the simulation setup considers channel fluctuation and congestion-based perturbations, we consider a network more "robust" than the another comparable network, when it exhibits a higher level of packet receipt. The distributions of network packet receipt rates are presented in the Figure 13. The distributions for network sizes 400 and 500 are presented in Appendix (8).

4.2.3 Motif structural redundancy and packet receipt

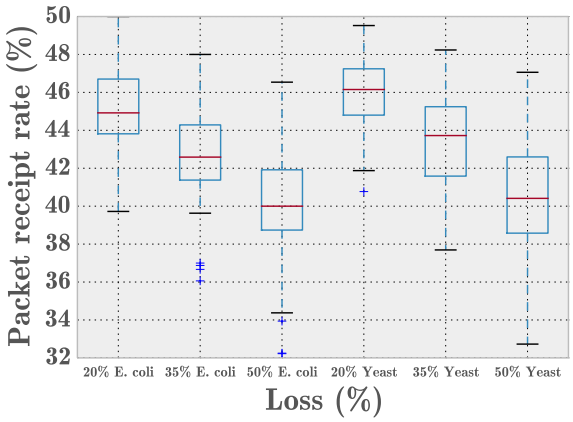
What is the the impact of structural redundancy, contributed by transcriptional motifs, on the information flow (packet transmission) through a complex network? In the context of the NS-2 framework, packets are successfully transmitted if those sent from a source node reach the sink (destination) node(s). That feed-forward loop transcriptional motifs (e.g. Fig. 14 (b)(1)) are hierarchical, and attenuate signal properties, such as response-time acceleration or delays [40], without any feedback loop, begs the question of whether they influence information transport at the more extensive network level. To examine this, we first tracked and identified all paths (node-hops) traveled by successfully received packets. We then used this history to identify all feed-forward loops that possess a nonempty intersection with these successful paths.



(a) $n = 100$ nodes.



(b) $n = 200$ nodes.



(c) $n = 300$ nodes.

Fig. 13.: Packet receipt rates (PRTs) for sampled transcriptional subnetworks of the bacterium *Escherichia coli* and *Saccharomyces cerevisiae* (labeled 'Yeast').

4.3 Support Vector Machine Modeling

Machine learning (ML) techniques can be used to discover and identify underlying patterns present in a given dataset. Currently, ML techniques are widely used for different purposes, such as to identify email spam, predicting election results, Internet search suggestions, targeted advertising, to name just a few. Among an army of techniques, support vector machine (SVM) is a supervised ML technique used for classification of data [20]. Our goal here is to first identify, and then to determine, which topological features of transcriptional networks best capture the behavior of a test network.

An SVM model identifies a *classifier* (boundary that separates data) which best classifies the given data. While linear classifier suits well in few instances, other instances may require non-linear separation boundaries. The implementation of such linear or non-linear boundaries in an SVM model is achieved using kernel functions. This classifier is often referred to as a hyperplane that separates instances belonging to different classes. The possible kernel functions include: linear, polynomial, radial basis function (RBF) and sigmoid. An SVM model predicts the target value of the test data given the features of test data.

An illustration of SVM dataset is shown in Figure 14(a). In SVM modeling, a dataset contains set of instances, and each instance is a combination of labels and features. The term 'label' is attributed to an output which describes a feature, which is a property of the dataset used. In addition, each feature is assigned a unique ID. For example, we employed ten datasets, which constitute five sampled subnetworks each from the transcriptional datasets for *Escherichia coli* and *Saccharomyces cerevisiae*. Each of these five datasets corresponds to a particular network size, as measured by the number of nodes, i.e. $n = 100, 200, 300, 400$, or 500. One hundred networks were

sampled from the source datasets for each size, and each such sampled subnetwork is an example of an 'instance'.

We used the Python programming language [63] and *scikit-learn* package [**scikit-learn**] to identify features and build SVM classification models. *scikit-learn* utilizes the popular ML libraries *libsvm* and *liblinear*. We follow the data preprocessing and model selection steps as prescribed by [21]. We perform data scaling after feature determination (Section 4.3.5) then perform grid search (Section 4.3.4) to identify best parameters to classify data. Our goal is two-fold: a) to build a classification model b) rank features. The proposed classification model will be used in the future to predict new data. Feature ranking is performed using analysis of variance F-test which does not use model created by SVM.

4.3.1 Assigning labels for SVM

As shown in Figure 24, packet receipt rates are calculated from each network using NS-2 from each network instance, and then a *k-means* clustering algorithm is employed to generate appropriate labels. k-means algorithm is applied to packet receipt rates (PRRs) as noted in Figure 14. The k-means algorithm partitions a number of points into clusters by first randomly assigning a center for each cluster; then, uses the 'distance' of each point to all cluster centers to determine which cluster to assign any given point. This process is iterated until the clusters are defined so as their 'centers' no longer change. Our two resultant vectors now are the label vector Y (100 rows $\times$1 column) and the corresponding feature vector X (100 rows $\times$16 columns). Each row in label vector Y corresponds to each row in feature vector X (Fig. 14(a)). The vectors X and Y together are termed as the dataset since it contains labels and features for a particular network size at a specific perturbation level.

4.3.2 Data pruning

A one-size-fits-all SVM model may not fully explain patterns within our datasets, such as statistical outliers of packet receipt from the NS-2 simulations, which become evident when clusters are identified using *k-means* clustering technique; because statistical outliers represent rare, large fluctuations, they may erroneously end up defining their own cluster. To avoid this problem, the dataset can be pruned by removing the labels and their corresponding data instances from the feature instances. Of course the best approach is to gather a maximum number of points to describe one network size, and this will be considered in future work. Consider the label vector Y with four clusters (IDs: $0, 1, 2, 3$) to be $\{1 : 37, 0 : 34, 3 : 28, 2 : 1\}$. Only one point belongs to cluster ID 2 and hence that point is discarded along with the corresponding feature instance vector. Now, the training and testing is performed on Y which is 99 rows $\times 1$ column and X which is 99 rows $\times 16$ columns. In this work, data was not pruned.

4.3.3 Training and testing

Nevertheless, the pruned data is used as training and testing sets for the machine learning models. Each dataset is split into 75% training and 25% testing sets. In order to avoid overfitting the data, 5-fold cross validation is used to randomize the 75/25 split into training/testing datasets. In a 5-fold cross validation test, the split is performed five different times; labels are stored in a vector, and corresponding feature instances are stored in another, different vector. Continuing the example stated in the Section 4.3.2, now the training set contains $\{1 : 27, 0 : 26, 3 : 21\}$ and the testing set contains $\{1 : 10, 0 : 8, 3 : 7\}$.

Table 3.: Grid search parameters identified using the cross validation method described in the text (20% perturbation).

Network size(s)	**Kernel**	**C**	**Gamma (γ)**	**Degree**
Yeast: 100, 500	RBF	100, 1	0.1, 2	-
Yeast: 200, 300, 400	Polynomial	1, 1000, 10	1, 1, 1	2, 1, 1
E. coli: 100, 200, 300, 400, 500	RBF	10, 10, 100, 1, 100	1, 0.1, 0.1, 2, 1	-

4.3.4 Parameter selection

A grid search is performed to identify the 'best' parameter set in which to build an SVM model. Grid search uses k-fold cross validation and builds a classifier for each set of parameters. Each classifier is then tested using the $F1$ *score*, which can be understood as a weighted average of precision and recall [49]. The set of parameters used are shown in Table 3. C is the regularization constant and γ is a kernel hyperparameter [1] used in non-linear kernel functions. Large C overfits the data (high cost for misclassification). Large γ in polynomial kernel ensures a smoother decision boundary.

4.3.5 Features

A machine learning technique uses underlying properties of the data to describe relationships between data instances, and these properties are referred to as features. For each instance of data, features are mapped to corresponding labels, which we describe below. Given a network of nodes and edges, $G(V, E)$, wherein V is the set

[1]Due to limited space the parameters are described here. 1, 10, 100, 1000 are used as C values for Linear, RBF, Polynomial kernels. The set of values 0.0001, 0.001, 0.01, 0.1, 1 and 2 are used as γ for RBF kernel. A γ value of 1 is used for polynomial kernel. 1, 2, 3, 4, 5 are used as *degree* values (applicable only to Polynomial kernel).

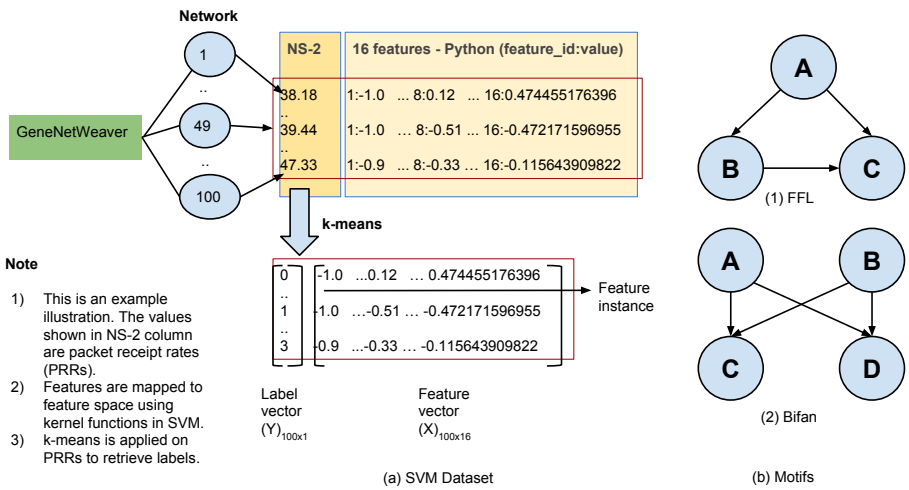


Fig. 14.: Illustration of (a) SVM Dataset for each network size, at specific perturbation level and (b)(1) FFL, (b)(2) bifan motifs respectively

of supporting vertices, and E is the set of edges linking those vertices. We define the following SVM features:

In what follows, features defined based on the network topology are given in sections 4.3.5.1 to 4.3.5.11, whereas features defined in terms of NS-2 simulation traces are given by sections 4.3.5.12 to 4.3.5.13. These latter features are referred to hereon as 'path-based features.'

In total, sixteen features are studied. All features/metrics are normalized to the interval $[-1, 1]$ to remove any artificial bias towards high-valued features. This can be carried out according to the following equation:

$$F_{js} = 2 \times \left(\frac{F_j - F_{min}}{F_{max} - F_{min}} \right) - 1, \tag{4.1}$$

wherein F is the set of features, F_{js} is the scaled jth feature value, F_j is the jth feature value, F_{max} and F_{min} are maximum and minimum values in F.

4.3.5.1 Network density

Network density (ND) is a measures of the number of edges in the network, $|E|$, against all possible edges, $|V|(|V|-1)$. Thus, it can be given by the following equation:

$$ND = \frac{|E|}{|V|(|V|-1)}. \tag{4.2}$$

4.3.5.2 Average shortest path

The average shortest path (ASP) of a network is the shortest of all path-lengths, $\min\{d(V_1, V_2)\}$, measured between any two network nodes V_1 and V_2. This metric captures the ability of two nodes to communicate information between them. For example, two adjacent nodes can be expected to communicate more frequently than two far-separated nodes in a noisy environment. We may compute this quantity according to the equation:

$$ASP = \sum_{V_1, V_2 \in V} \frac{\min\{d(V_1, V_2)\}}{|V|(|V|-1)}. \tag{4.3}$$

4.3.5.3 Degree centrality

Degree centrality of a node is defined as the number of edges incident to the node. Thus, it provides a measure reception to others within a network. In order to identify the impact of genes, which are regulated by transcription factor proteins in a transcriptional network, the collective average degree centrality of genes (ADCG) is considered as a feature, along with average degree centrality of the network (ADC). The degree centrality of a node can be determined as follows:

$$n_{dc} = \frac{deg(n)}{|V|-1} \tag{4.4}$$

wherein n_{dc} is the degree centrality of node n and $deg(n)$ is the degree of node n.

4.3.5.4 Transcription factor percentage

Transcription factor percentage (TFP) provides a measure of the fraction of networked nodes that serve as transcription factors which regulate genes. This can be calculated as follows:

$$TFP = \frac{|V_{TF}|}{|V|}, \tag{4.5}$$

wherein $|V_{TF}|$ is the number of sum-total of transcription factor nodes within the network.

4.3.5.5 Genes percentage

In complement to TFP metric, Eq. 4.5, we define the genes percentage (GP) as the fraction of networked that can be identified as genes. This quantity can be calculated with the equation:

$$GP = \frac{|V_G|}{|V|}, \tag{4.6}$$

wherein, $|V_G|$ is the number of gene nodes.

4.3.5.6 Source to sink edge percentage

Larger networks are more likely to support links that directly connect source to sinks within the network, facilitating information flow. Thus, we propose a metric that quantifies this property: the source to sink edge percentage (SSEP), which we define as the fraction of direct edges, $|E_{SS}|$, from source nodes to sink nodes compared to the total number of edges in the network:

$$SSEP = \frac{|E_{SS}|}{|E|}. \tag{4.7}$$

4.3.5.7 FFL abundance

Feed-forwad loop abundance (FFLD) is the ratio of total edges in the network that intersect with edges from at least one feed-forward loop to the total edges in the network. Thus, it can be calculated with the equation:

$$FFLD = \frac{|E_{FFL}|}{|E|}, \tag{4.8}$$

where E_{FFL} is the number of edges that participate in feed-forward loop transcriptional motifs.

4.3.5.8 FFLDED

Figure 14(b)(1) illustrates a feed-forward loop transcriptional motif, which is hierarchical, but composed of two regulatory paths. The first is a 'direct' linkage from nodes A to C, whereas an 'indirect' path accounts for regulation of node C through a node B waypoint. Here, the feed-forward loop direct-edge density (FFLDED) is the ratio of feed-forward loop direct edges, $|E_{FFLDE}|$, to the total edges in the network, and may be calculated using the equation:

$$FFLDED = \frac{|E_{FFLDE}|}{|E|}. \tag{4.9}$$

Note that the FFLDED may be > 1, because several feed-forward loops may utilize the same direct-edge linkage.

4.3.5.9 FFLSSPD

The feed-forward loop source to sink edge density (FFLSSPD), is the fraction of direct source-sink edges that are also part of a feed-forward loop to the total number of source-to-sink edges in the network. This metric decouples the influence of feed-

forward loops from all other source-to-sink edges in the network.

4.3.5.10 FFLDEP

The FFLDED metric above (Eq. 4.9, accounts for the fraction of direct-edge feed-forward loop links present within the network topology. However, a single linkage may potentially appear more than once if it is 'shared' among two or more feed-forward loops. We define a separate measure that ignores multiple copies of any single link, which can be calculated as follows:

$$FFLDEP = \frac{|E_{FFLDE}|}{|E|}, \tag{4.10}$$

wherein $|E_{FFLDE}|$ is the number of unique direct-edges in for feed-forward loop transcriptional motifs embedded within the network.

4.3.5.11 FFLIDEP

Indirect FFL edge percentage (FFLIDEP) is the ratio of the number of unique feed-forward loop indirect edges to the total number of sequential, two-step paths in the network. Thus, it is similar to the FFLDED metric above (Eq. 4.10), but measured against the indirect edge of the feed-forward loop. This can be calculated with the equation:

$$FFLIDEP = \frac{|E_{FFLIDE}|}{|E_{TEP}|}, \tag{4.11}$$

wherein $|E_{FFLIDE}|$ is the number of indirect edges (two-step paths) in feed-forward loop motifs, and $|E_{TEP}|$ is the total number of sequential two-edge paths present in the network proper.

4.3.5.12 Direct-edge trace participation

Each NS-2 simulation results in a set of 'traces' that map packet-transport histories for packets sent and received successfully from source to sink nodes. In a similar concept to that of Eq. 4.9, but accounting for packet trace history, we measure the ratio of the number of unique feed-forward loop direct edges that participate in successful packet paths to the number of unique FFL direct edges, termed FFLDSPATH.

Another related feature can be defined similarly to FFLDSPATH: if we allow for duplication of feed-forward loop direct-edges, then we term this count FFLDOSPATH. That is, this metric allows for feed-forward loop direct edges to participate multiple times in successful packet delivery.

4.3.5.13 Indirect-edge trace participation

Finally, we measure the ratio of the number of unique active FFL indirect edges that participate in successful packet trace histories to the number of unique feed-forward loop indirect edges. This metric is termed FFLIDSPATH.

Similar to above, we allow for the multiple counting of a single feed-forward loop indirect path in the contribution to successful packet trace history. This metric is termed FFLIDOSPATH. That is, feed-forward loop indirect edges can be leveraged more than once to successfully deliver a packet.

4.3.6 Feature ranking

The identified features are ranked using the analysis of variance (ANOVA) F-value metric. This metric compares the inter-class variance to intra-class variance [**scikit-learn**]. A higher F-value denotes higher significance of a feature. F-value captures feature significance individually but mutual feature dependence cannot be

determined. We intend to use different metrics in the future work.

4.4 Results

4.4.1 Packet receipt rates using transcriptional network topologies

Figure 13 illustrates the distribution of packet receipt rates (PRTs) for representative subnetworks sampled from *Escherichia coli* and *Saccharomyces cerevisiae*, across three different loss models (20%, 35% and 50%). Outliers in the dataset are points that do not occur in the range of top and bottom whiskers and are identified by $+$.

Generally, all simulated packet-transport scenarios exhibited packet receipt rates that decreased, on average, with an increase in the loss model. This trend persisted across subnetworks sampled from both *E. coli* and *S. cerevisiae* (i.e. 'Yeast' networks), of all sizes, but the smaller subnetworks ($n = 100$) exhibited the most variability. That larger networks were less efficient should be expected: the number of possible paths between two nodes increases as the network increases. Because packets may 'disappear' during any given hop between nodes, the increase in total edges should correlate with a subsequent decrease in received packets, independent of the global network topology.

4.4.2 Feature ranking in transcriptional networks

Perturbation in a transcriptional network can either be external or internal. In the view of NS-2 simulation framework, channel noise and congestion based packet drops account for internal perturbations. As mentioned above, fluctuation in packet loss (%) is considered as a perturbation/stressor to the information flow. This channel loss stressor is used using the SVM models to explore the significance of transcriptional

motifs on structural redundancy and packet receipt rates.

4.4.2.1 Top-ranking features

Fifteen different SVM models, one for each pair of network size and perturbation level, are used to select features/metrics for one specific type of transcriptional network. Let us examine the feature selection in *E. coli* networks for one of the fifteen SVM model instances. For each network size, the top five features are selected, according to the criterion that each the most 'influential' features should occur at least three times in the top five features as scored across different network sizes. For *Escherichia coli* networks, this top-ranking set is given by the features: TFP, FFLIDOSPATH, ASP, FFLIDEP, ADCG, FFLD (Fig. 15a). Similarly, features so identified from the *Saccharomyces cerevisiae* networks are: FFLIDEP, TFP, FFLD, GP, FFLIDOSPATH (Fig. 15b). All influential features identified from the SVM models in terms of packet receipt rates relate to the feed-forward loop subnetworks.

4.4.2.2 Feature stability at different perturbation levels

As a preliminary experiment, we tested the prevalence of transcriptional network features at different noise perturbation levels. Here, our intention is observe if structural or dynamic features prevail in feature significance. The result of this on *E. coli* networks is shown in Figure 15a[2] and on Yeast networks is shown in Figure 15b. FFLIDEP ranks consistently higher in most cases (except at network size 100) than other features. Similarly, FFLD and GP rank in the top two or three at different network sizes. An interesting observation is that three (FFLIDEP, FFLD, FFLIDOSPATH) out of five top ranked features are related to FFL motifs.

[2]For the figure to be legible, X and Y labels are displayed only once. This is done for Figures 15a - 18.

4.4.2.3 Feature ranking variation across different network sizes

We now observe if the relative importance of features varied across different network sizes. From Figure 16a, it can be seen in *E. coli* networks that TFP ranks consistently stable in most cases in 35% and 50% perturbation levels (except at network size 300). FFLIDOSPATH, FFLD and FFLIDEP rank higher in some instances. Figure 16b shows the relative importance of features in Yeast networks. Here, FFLIDEP is relatively stable across different network sizes compared to other features. FFLD along with GP seems to be stable at certain instances but not conclusively overall. A combination of conventional metrics such as GP and motif-derived features can be used to engineer special networks which can ensure stability across different perturbation levels.

4.4.2.4 Comparison of FFL based features

Identifying features that are significant to network robustness will be substantial to design specially engineered networks that are *functionally* robust and can withstand perturbations. The results from the above two studies give us an opportunity to observe variation of FFL based features only instead of the top five identified features. A general trend can be observed from Figure 17 that FFL-based features have higher significance (based on normalized ANOVA F-value) from network sizes 300 and above. Second inference from Figure 17 is that FFLIDEP is ranked first among the six FFL based features in certain instances (100, 200, 300 and one instance in 400, 500 network sizes each). Figure 18 shows the ranking for Yeast networks. FFLIDEP ranks the highest for all network sizes and at all perturbation levels. Correlation between FFLDSPATH and FFLDOSPATH (derived from FFLDSPATH) is not always proportional suggesting that there is more to FFL participation and the number of

successful FFL direct path contribution. The position of FFL might also be critical for prevalence of certain features. FFLDEP, FFLIDEP and FFLIDOSPATH consistently rank as the top three features at different perturbation levels. This directly reveals the importance of the percentage of FFL direct edges present in the network and the number of times those edges were used in successful packet transmissions.

4.5 Discussion and conclusions

A key aspect before identifying and ranking features is mapping packet receipt rates to labels using *k-means* clustering algorithm. Choosing the optimal cluster size is crucial for creating labels. If one single point is equidistant from all different clusters, it will eventually remain in its own cluster. This problem can be addressed by gathering as many instances as possible for a given network size. Training to testing data set split ratio is critical for creating a classification model. Selecting a high training set percentage will overfit the data. Another challenging aspect is the data loss due to pruning (as explained in Section 4.3.2). Feature ranking can potentially be influenced by inappropriate data pruning. Using sufficient number of data instances can address this problem.

The design of future engineered systems may be inspired by naturally occurring robust systems, and a knowledge of features that exploit structural properties of transcriptional motifs are beneficial to these design efforts, especially if they vary depending on the network size. Wireless sensor networks are just one application for such systems, wherein developing adaptive routing mechanisms for information transport is crucial for efficient communication performance.

We studied transcriptional networks of the model bacterium *Escherichia coli* and the common baker's yeast *Saccharomyces cerevisiae* to identify system-defining features based on topological considerations of the these networks, but also based on

dynamical properties of information flow across them. To this effect we used the NS-2 based discrete event simulation framework, and support vector machine learning methods from the field of machine learning, to recognize and identify underlying patterns in these transcriptional subnetworks. We discovered that feed-forward loop based metrics consistently outperformed traditional metrics such as network density, average shortest path, and degree centrality based measures. Whether other transcriptional motifs contribute to improved function remains a focus of future work in this area. Nevertheless, it remains to be seen how far topological considerations alone can be pushed to improve information-flow properties in engineered networks, because biology employs many other mechanisms that feed off of the regulating topology, such as protein conformation states, association or dissociation events (e.g. dimerization), complexation states, or post-transcriptional and post-translational modification of protein activity, such as the phosphorylation state.

4.6 Biorobust

In order to share our work with the research community, we created BioRobust. BioRobust is an online framework to quantify biological network robustness. This is complementary to the approaches presented primarily in Chapter 4.

Several technologies were used in the process to create BioRobust. Django and Python are used to power the web framework. Bootstrap3 library developed for Django ([68]) is used for user interface. The site is hosted on a Unix server and can be accessed at http://bnet.egr.vcu.edu:8000/.

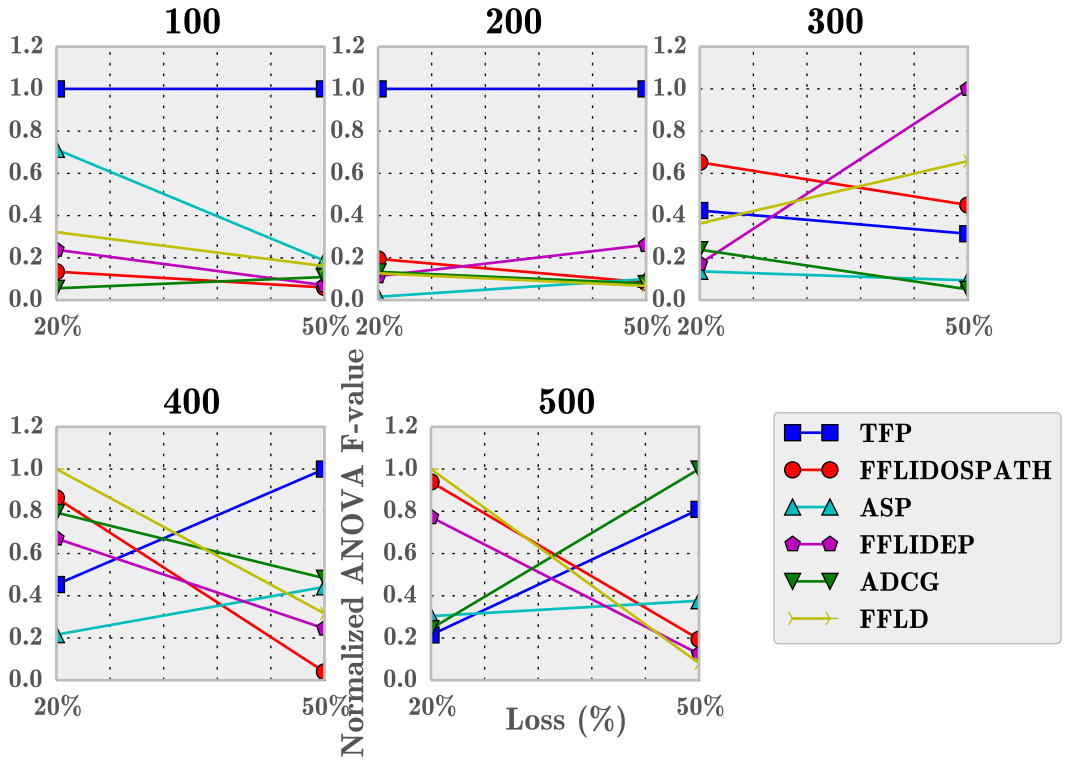
Users can submit their choice of files for robustness analysis. The pipeline of the analysis process is showcased in Figure 19 below.

The user is requested for their email address using which a directory is created and the submitted file is uploaded along with the parameters selected. In order to

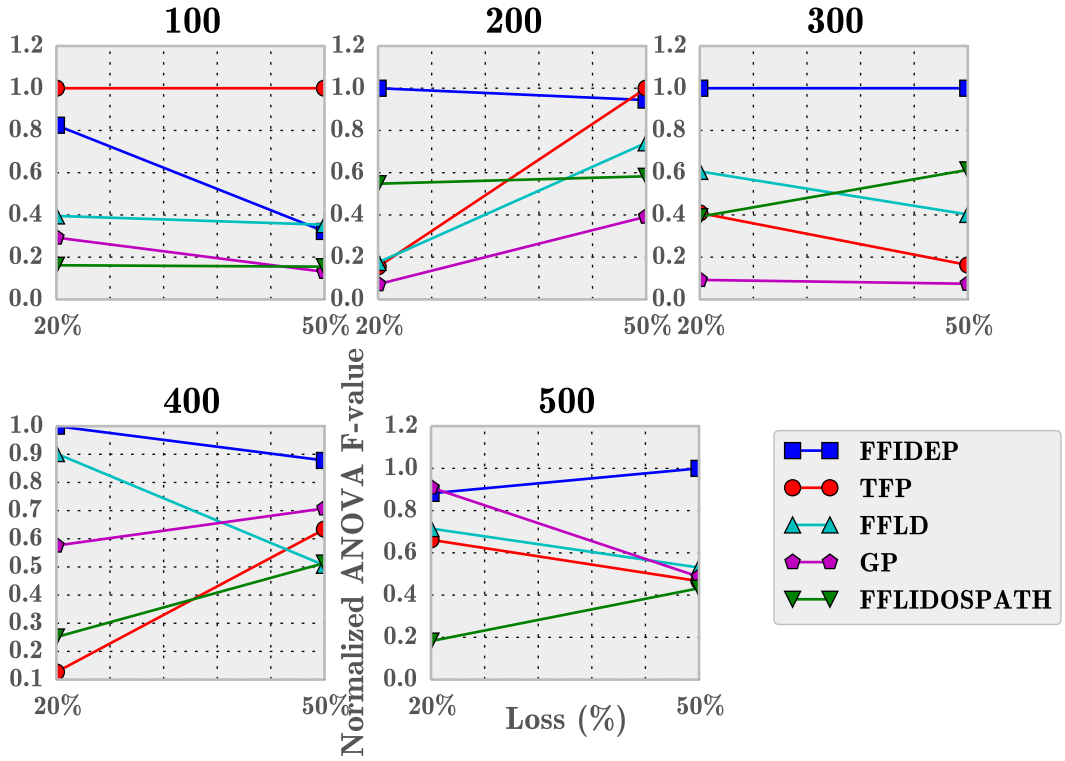
avoid naming conflicts user email address is combined with a random number for directory creation. Users can, along with the network file, mention direction of the network and category of the network as shown in Figure 20.

A Unix script checks for user submitted files every fifteen minutes and triggers the process pipeline upon observing the newly created directory. Network simulations to determine network robustness are then executed followed by examining the trace files generated by the simulations. These trace files indicate us the paths taken by packets during information transmission among the participating nodes. Currently, the software supports modules (subgraphs) extracted from the transcriptional regulatory networks of *E. coli* and baker's yeast. Machine learning models are currently trained for the network sizes 100, 200, 300, 400, and 500. Here, the number of nodes in the network represent the network size. Results determined using the networks that belong to a different category (than E.coli or Yeast) are not applicable for research insights as the models developed do not depend on the underlying network category. Once the file(s) are submitted and the analysis is complete, users are notified when the results are available for viewing (Figure 21).

BioRobust also presents the data used for analyzing biological networks. The models trained for predicting the network robustness are developed using random forest regression strategy. To this end, a total of hundred runs of models were executed to negate the fluctuations due to randomness in the random forest algorithm. As network size increases, the time taken to analyze the submitted network increases proportionally. Support can be extended to other types of networks depending upon user interests.

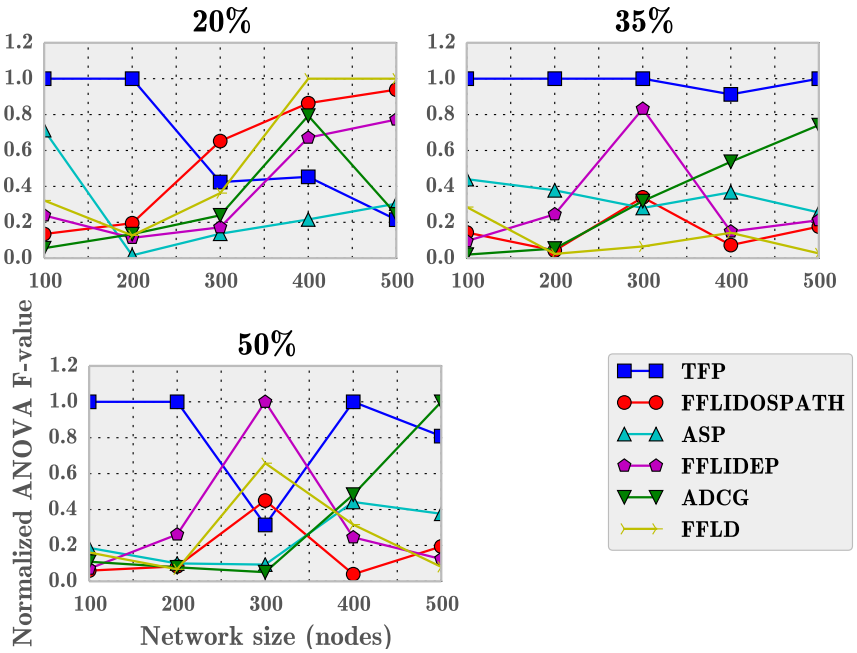


(a)

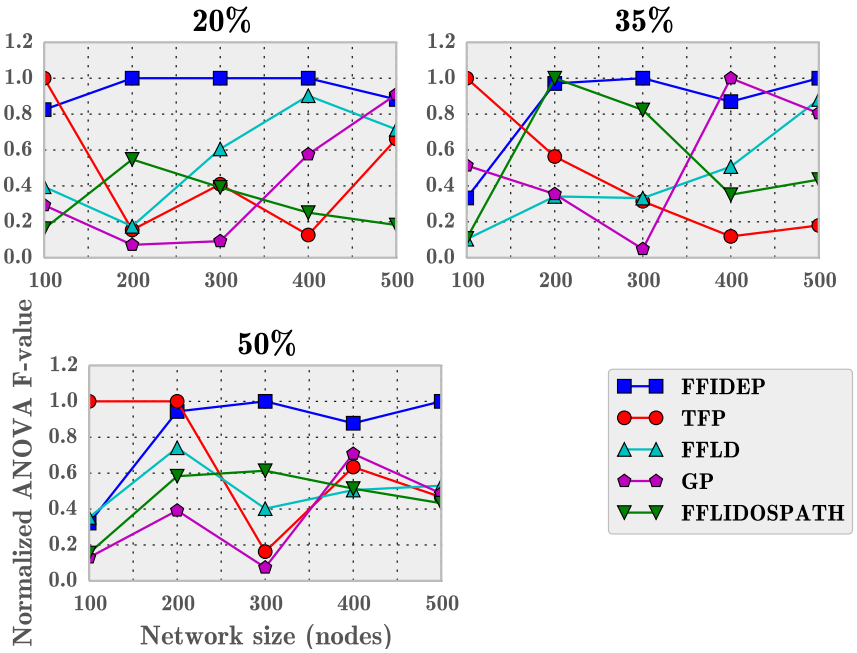


(b)

Fig. 15.: Variation of top 5 features in each *Escherichia coli* network (panel (a)) and *Saccharomyces cerevisiae* (panel (b)) networks, at losses 20% and 50% (Sizes = 100,



(a)



(b)

Fig. 16.: Variation in normalized ANOVA F-values for the top 5 features in each *Escherichia coli* network (panel (a)) and *Saccharomyces cerevisiae* (panel (b)) networks, at losses 20% and 50% (Sizes = 100, 200, 300, 400, 500).

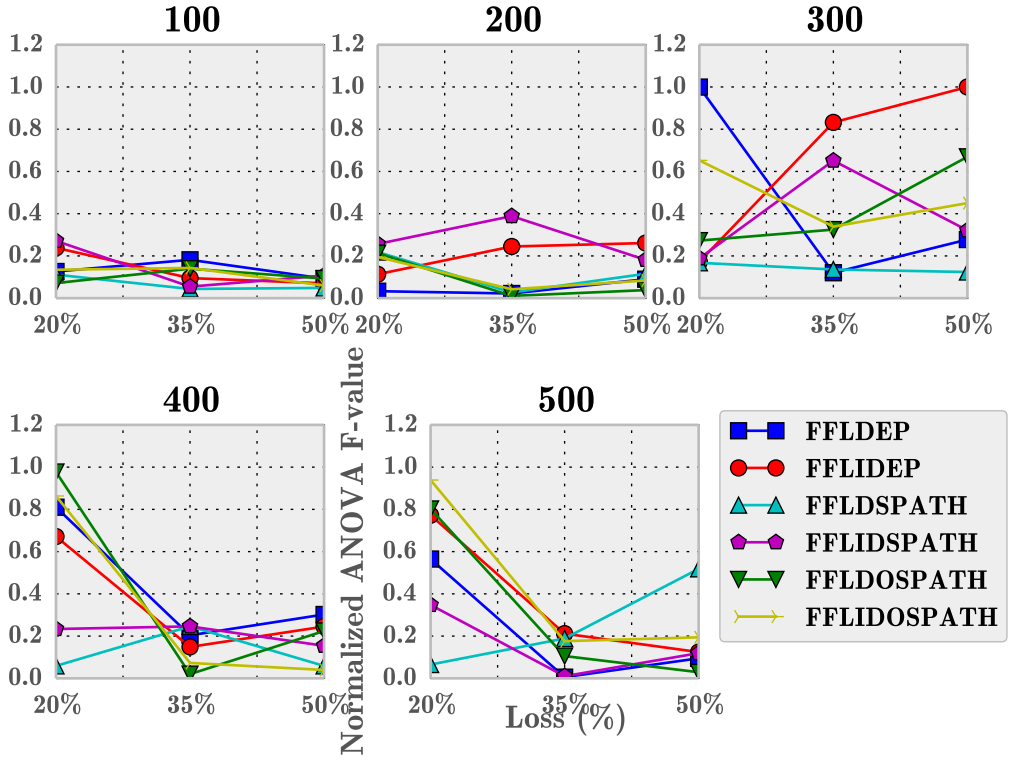


Fig. 17.: Variation of FFL participating direct and indirect edge-based features at 20%, 35% and 50% loss each for different E. coli networks (Sizes = 100, 200, 300, 400, 500).

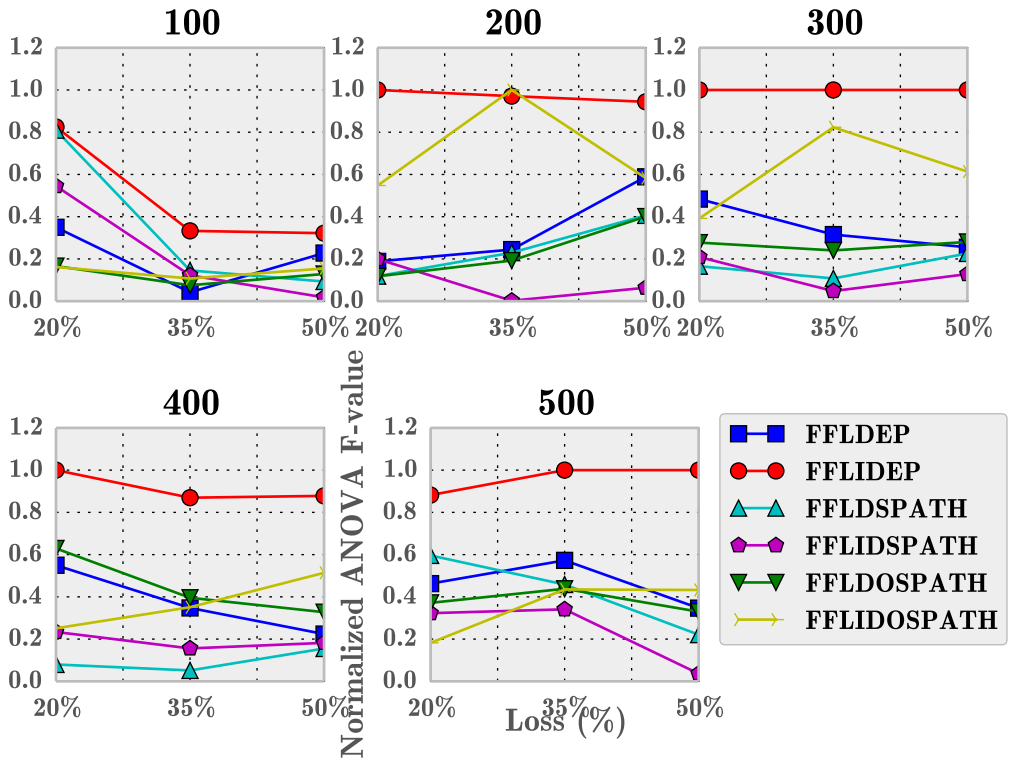


Fig. 18.: Variation of FFL participating direct and indirect edge-based features at 20%, 35% and 50% loss each for different Yeast networks (Sizes = 100, 200, 300, 400, 500).

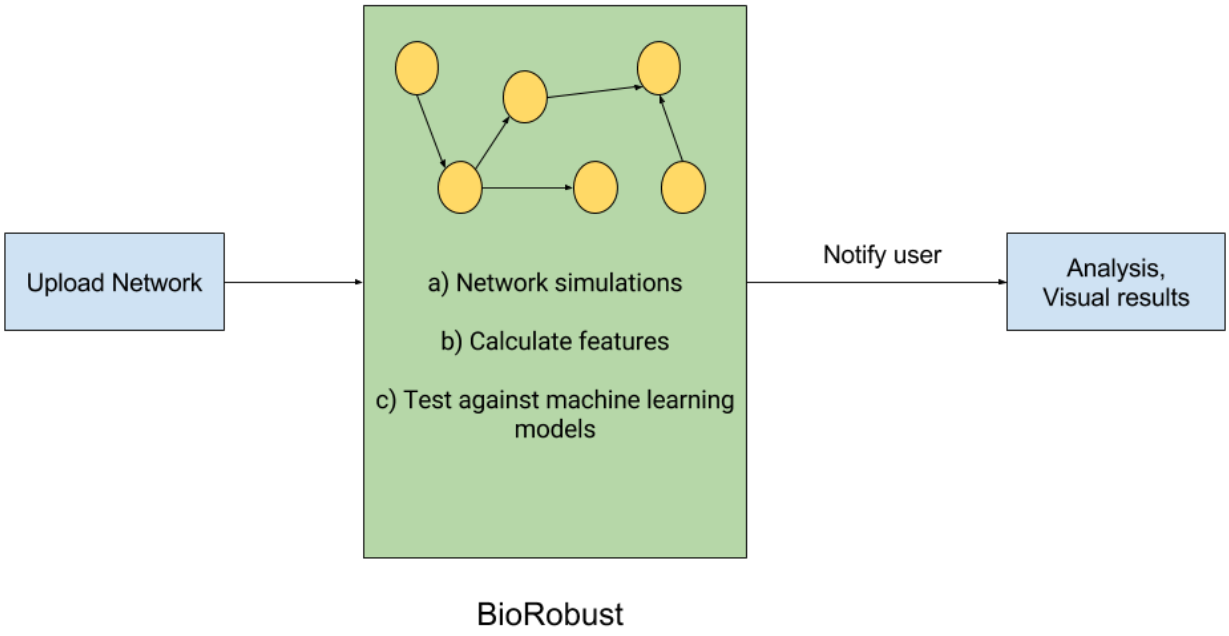


Fig. 19.: BioRobust flowchart

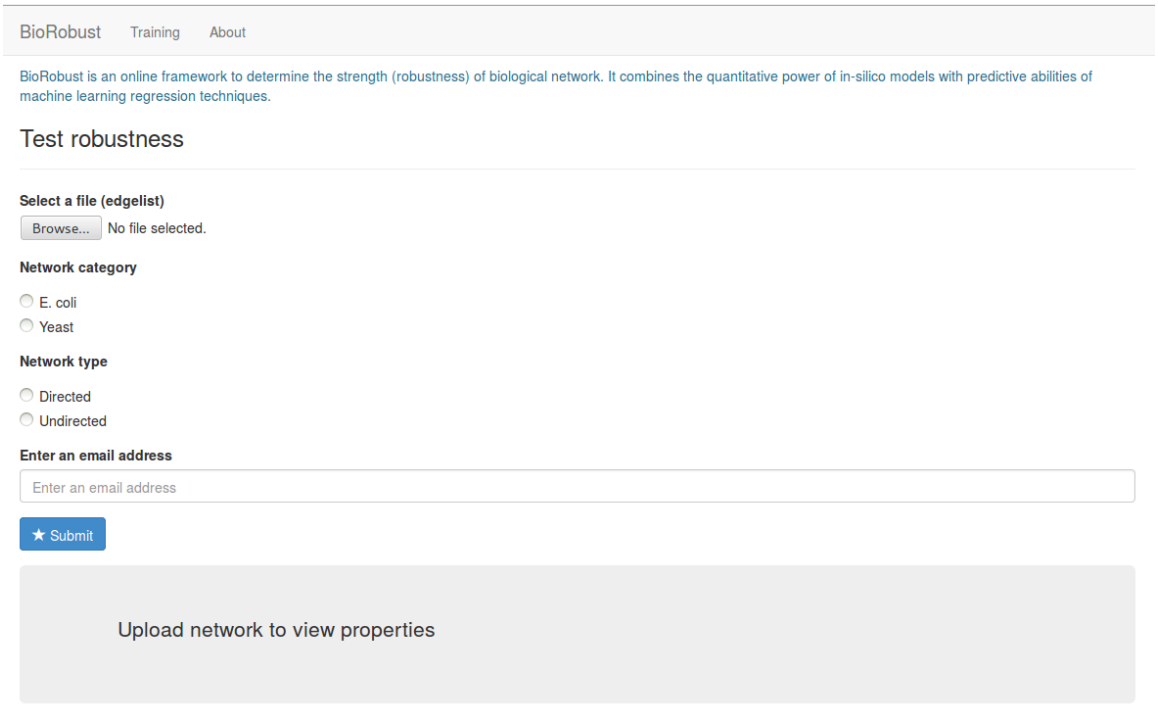


Fig. 20.: BioRobust prototype - Select network for analysis

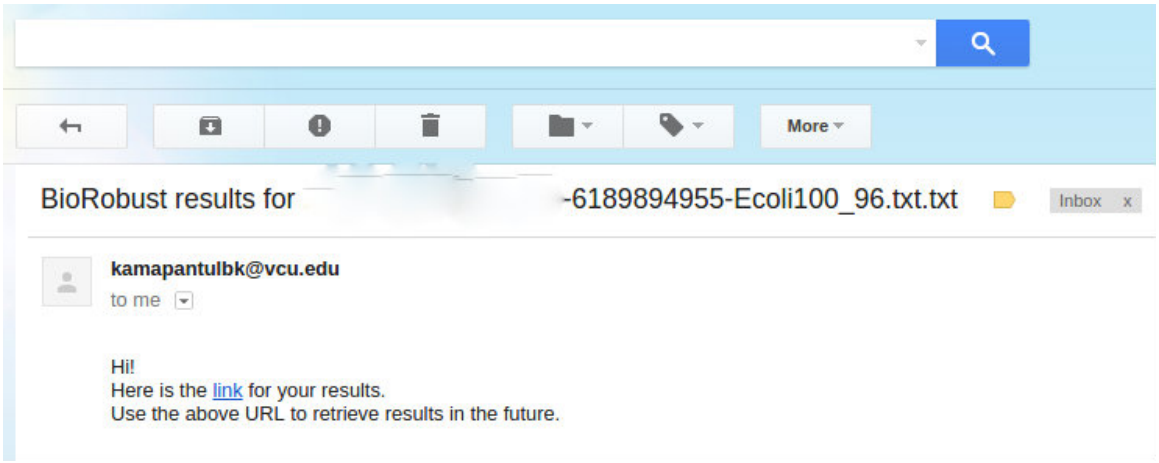


Fig. 21.: BioRobust prototype - User notification of results (Email blurred).

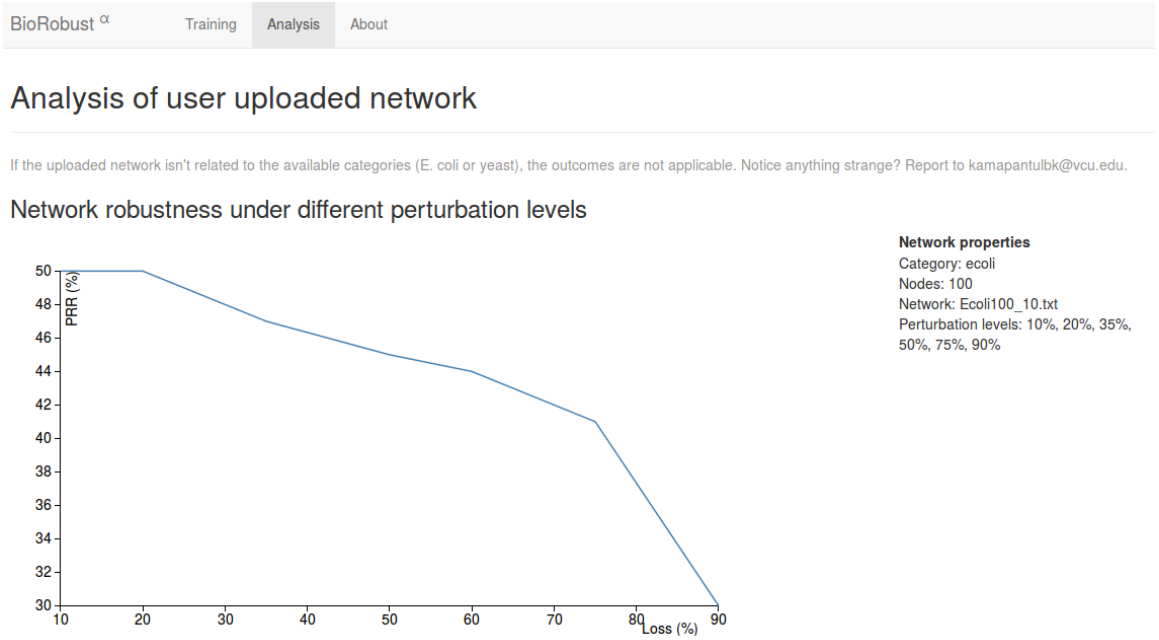


Fig. 22.: BioRobust prototype - Robustness analysis of the uploaded network across different perturbation levels.

Fig. 23.: BioRobust prototype - Feature importance of the uploaded network across different perturbation levels.

CHAPTER 5

ABUNDANCE OF CONNECTED MOTIFS IN TRANSCRIPTIONAL NETWORKS

5.1 Introduction

Motifs are often attributed to be one of the reasons for *robust* biological systems. A repetitive structure that occurs with a higher statistical significance in real networks than in random networks is termed to be a motif. In the past, researchers have identified feed-forward loop (FFL) motif to be an important motif in terms of abundance [43]. Further, functional significance such as response time speed-up and slow down has been attributed to FFL motif [40]. FFL structure is intriguing not only for its role in biological functionality but structurally as well Figure 24(b). It offers two ways of regulating the gene node (C) via two different transcription factor (A, B) nodes. In communication scenario, this becomes crucial when there is a network failure but information still needs to be transmitted. It is likely that the presence of higher FFL motifs will lead to better information transmission. In this work, we take a step further to study the connectivity between FFL motifs.

For the first time, this work aims to study the importance of the abundance of connected motifs. We use discrete event simulations and machine learning techniques to create a model, train and *learn* the feature data and predict *robust* behavior of biological network topologies. Discrete event simulations assist in modeling dynamic behavior of network interactions (information flow among the nodes in a network) under controlled conditions such as channel noise and congestion-based information loss. We assume that features in a biological network can be ranked. Does higher abun-

dance of a connected motif pattern mean a *robust* network? Which of the considered network features contribute to *robustness*? Which machine learning model can accurately predict the *robust* behavior of biological network topologies? We explore these questions in the following sections. Answering these questions will reveal insights to the working of robust biological network topologies leading us to engineer specialized networks which are resilient under heavy perturbations. Section 5.2 presents the methodology followed in this work. The definition of *robustness* varies from context to context. The metrics studied by researchers are predominantly static in nature [6, 51] as they do not consider the dynamic information flow within the network. [6] provides an in-depth review of existing metrics to measure *robustness*. None of the metrics consider features based on motifs or even connected motifs. *Robustness*, in our work, is measured in the aspect of successful information transmission as modeled by a discrete event network simulator. To this effect, we define network robustness as the ratio between the total number of packets received at the sink nodes to the total number of packets sent from the source nodes. We term this metric as packet receival rate. Packet receival rate is a dynamic metric as it models the network behavior at different perturbed conditions. This experiment setup has been detailed in our prior work and can be noticed in [24].

5.2 Methodology

The methodology followed in this work is illustrated in Figure 24. Subnetworks extracted (Section 5.2.2) from *E. coli* transcriptional regulatory network are passed to network simulator platform NS-2 (Section 5.2.3) to generate packet receipt rates and feature values are determined using Python programming language [63]. As a standard practice, features are scaled between 0 and 1. Section 5.2.4 describes Data processing followed in this experiment is described in Section 5.3.1. After processing

the data in the correct format (as mentioned in the Step 1 in Figure 24) random forest regression machine learning technique is applied for feature ranking, and output prediction. Mean square error metric is used to determine the optimal number of estimators (a key measurement used to estimate random forests) number (described in Section 5.3.2). Before feature ranking is actually performed, we perform feature selection which is a process to reduce feature set (from a thirty eight feature set). Features are ranked using feature importances (a technique used to determine feature significance in regression trees). Section 5.3.2 details the parameters used for creating random forests regression models followed by the performance of vertex-shared motif features.

5.2.1 Contributions

The major contributions of this work are as follows:

1. Define vertex-shared motifs which are potentially responsible for biological functionalities.

2. Using random forests regression to select important biological network characteristics.

5.2.2 Transcriptional subnetworks

Escherichia Coli and Saccharomyces cerevisiae are considered to be model organisms in the biological networks research community. For this work, we extract transcriptional subnetworks from *Escherichia Coli* to understand biological network characteristics and motif interactions from a structural perspective. To this effect, subnetworks of different sizes are considered: 100, 200, 300, 400, and 500 (size represents the number of nodes in a network). For each size, 1000 transcriptional sub-

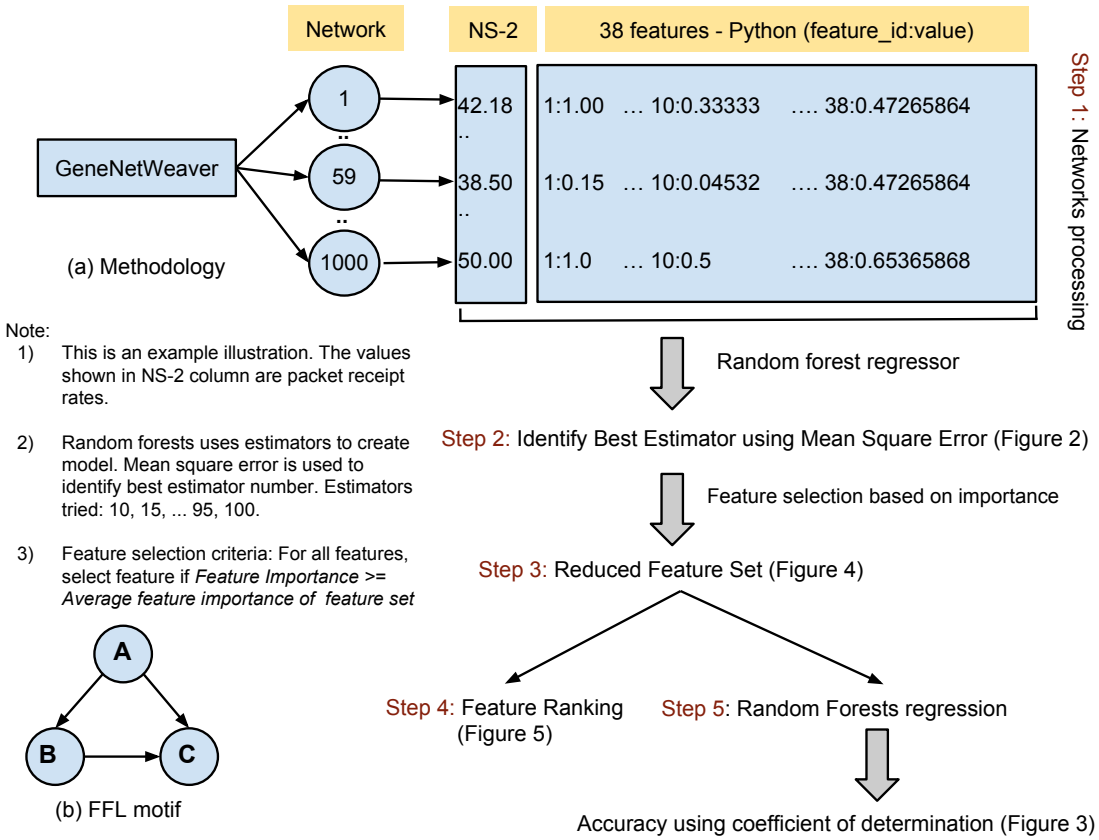


Fig. 24.: (a) Step-by-step methodology (b) FFL motif

networks are extracted using GeneNetWeaver software [57]. During subnetwork extraction, GeneNetWeaver retains critical biological characteristics such as modularity. Specifically, these modules are responsible for distinct biological functionalities. Direction of the edges within these networks is retained as it captures regulation information of genes by transcription factors. Networks that are disconnected are not considered for further analysis. Self-edges (node with edges directed towards itself) in each network are discarded and the remaining network is reconstructed. This step pruned the dataset to 947, 943, 957, 932, and 941 networks for 100, 200, 300, 400, and 500 [1] network sizes respectively. This dataset is then used to explore network dynamics in two ways: a) model interactions using NS-2 (Section 5.2.3) and b) determine structural features from a static and dynamic perspective (Section 5.2.4 and Section 5.2.5). We study the significance of these features using machine learning techniques, specifically using regression modeling. This helps us identify the variation in feature importance from one network size to the other and under several lossy conditions.

5.2.3 Modeling network dynamics using NS-2

Network simulator platform, NS-2, allows researchers to explore the network characteristics. Previously, we mapped the problem of information flow in a biological network to that in a wireless sensor network [16, 29, 27]. This setup helps us understand the characteristics of biological networks uniquely using a framework used for wireless sensor networks. Following which, we established NS-2 as a robustness framework for biological networks. In a NS-2 network simulation, information is transmitted across the network via nodes and edges. Each node sends information in terms of packets across its outgoing edges and these packets are collected at sink

[1] 871 networks were used for 500 network size at 10% loss. 941 networks were used for all other loss scenarios for 500 network size.

nodes. Transcription factors (also considered as source nodes here) and genes (also considered as sink nodes here) are both represented as nodes and the interactions among them are represented via edges. Packet transmission in each network is studied at various loss models: 10%, 20%, 35%, 50%, 60%, 75% and 90%. Packet receipt rate in the network is measured as the percentage of the number of packets received at sink nodes to the number of packets sent by all source nodes. Networks with higher packet receipt rate are considered to be more *robust*. Packet receipt rates of the networks range in between 0 (least *robust*) and 100 (most *robust*). Source nodes are considered to transmit or forward information (through packets) and sink nodes only receive the information. This situation is similar to a transcription network where gene is regulated (receiving information) by transcription factor(s).

5.2.4 Structural features

In order to understand the features contributing to *higher network robustness*, we studied several network characteristics. While some of these characteristics such as average shortest path, network density, and betweenness centrality have been explored by researchers under the context of robust networks, our definition of *what robust is* places emphasis on the study of network dynamics. In our earlier work, we identified fifteen different network features and ranked them using unsupervised learning techniques [25], [24]. These features include static characteristics such as average shortest path, network density, degree centrality and dynamic characteristics such as patterns derived from FFL-based direct and indirect paths [2]. These dynamic characteristics are derived after looking at the information flow using NS-2 simulation

[2]Consider an FFL ABC where C is regulated directly by A and indirectly by A via B. Here, the edge $A-C$ is considered to be a direct FFL edge and edge $A-B-C$ is considered indirect FFL edge

platform. This helps us identify the paths that were heavily used to transmit information and if these paths are related to FFL motifs. Some of the features use specific terminology from information communication theory (such as packet transmission). The order of the features studied in this work is as follows: 1) network density, 2) average shortest path, 3) average degree centrality of the network, 4) transcription factors percentage, 5) genes percentage, 6) percentage of source to sink edges, 7) abundance of direct FFL motif edges, 8) abundance of indirect FFL motif edges, 9) percentage of FFL direct edges that contribute to successful packet transmission, 10) percentage of FFL indirect edges that contribute to successful packet transmission, 11, 12) number of direct and indirect FFL edges compared to the total successful (that contribute to successful packet transmission) direct and indirect edge paths in the network, 13) percentage of total edges in the network that participate in FFLs, 14) percentage of total edges that are actually FFL direct edges, 15) percentage of FFL direct edges that are source to sink edges. While our earlier work focused on identifying the impact of FFL, this work is focused on determining the impact of two FFLs that are connected. To this effect, we defined twenty three different connected FFL features that capture the abundance of connected FFL structures which are described in the following section. In total, we study thirty eight features to model the regression predictor. Hereafter, we refer to the connected feed-forward loop motifs as vertex-shared motifs.

5.2.5 Vertex-shared motif connectivity

It has been argued that interactions among modules are responsible for specific functionality in biological networks [12]. This is a deviation from another standpoint which states that the abundance of some structural patterns contributes to network robustness. While it is imaginable for both views to be correct, here we explore the

structural role of specific modules in network robustness. Modules are essentially connected motifs at work. Here, we explore the vertex-shared feed-forward loop motifs for their structural role in attaining biological network robustness. In order to understand the significance of connected motifs, we first identified all possible ways two feed-forward loop motifs could be connected. Following the identification, we determined the abundance of each pattern in the above mentioned transcriptional networks. The motif patterns can be divided into three categories first of which is *bow-tie* where one vertex is shared between two FFLs, second being *rhombus* where two vertexes are shared between two FFLs and third category being *bi-triangle* where all three vertexes are shared by two FFLs. All these patterns along with their respective abundance values are tabulated in Tables 4 and 5. Out of eighteen possible rhombus patterned motifs, there are six instances (RH-1/RH-8, RH-3/RH-14, RH-4/RH-11, RH-6/RH-17, RH-9/RH-13, RH-12/RH-16) where two patterns are found to be structurally isomorphic. All the isomorphic structures are shown in Table 6.

5.3 Random forest regression

Machine learning techniques prove quite useful in identifying significant features among a list of several features. Different strategies are employed for this task of significant feature identification. To perform machine learning tasks, we use the widely recognized *scikit* [49] module in Python. The aggregation of features defined in Section 5.2.4 and Section 5.2.5 combine to a total of thirty eight features. Abundance of connected motifs does not always contribute to *robust* network behavior. Data for connected motif abundance for different network sizes is suppressed here due to space considerations but provided in Section **??**. The test for the correlation of feature abundance with robustness is performed in Section 5.3.4.

5.3.1 Data

Data is constructed similar to the procedure followed in our earlier work [24]. Each network is represented as a combination of feature values, feature ids and output labels. The output labels are determined using NS-2. In total, thirty eight features are studied in this experiment. These include the twenty three vertex-shared motif features introduced earlier apart from the fifteen features presented in [24]. As suggested in [21], we scale each feature between 0 and 1 for all the samples considered to create a model. Each network is represented as a combination of output labels and thirty eight network characteristics. This combination is known as a feature instance, in machine learning terminology. The results from NS-2 are used as output labels and the corresponding features are calculated using *networkX* [58] module in *Python* programming language. In our previous work [25, 24], we considered the problem of ranking features to be an unsupervised one and used ANOVA [3] F-value to determine the significant features. But here, we consider the problem to be a supervised one and retained the output labels (range between 0 and 100) as floating points. In order to use classification techniques, one would have to group the output labels into bins which would mask the real data. Regression techniques are best suited for continuous data as output labels to predict new data. In order to avoid points that are equidistant from all the clusters (as noted in [24]), we increased the sample size for each network size from 100 to 1000 networks. By treating the problem as supervised instead of unsupervised one, we further take advantage of the output labels from NS-2. Further, we introduce feature selection here an improvement from our earlier work where the entire feature set was used to rank features. Before creating regression model, data is split into training and testing data in 75:25 ratio. Data split step is

[3] analysis of variance

a common practice in machine learning tasks to 'train' the model on training data during which the model 'learns' the data and testing is performed on the test data. The accuracy of regression models presented in Figure 26 is based on testing of model created on the test data of all the 38 feature set.

5.3.2 Regression modeling

Firstly, network characteristics that are understood to capture the network robustness are defined. In our experiment, we have considered two scenarios, first one with a total of thirty eight features are considered in order to capture the network dynamics, and in second case twenty three features formed by the connected feed-forward loop motifs. However, before calculating an estimator that can be used to predict the performance of new network data, features need to be pruned. Some features might be correlated with each other and some might display higher variance than the rest. We considered different feature selection methods to achieve the need of feature pruning. Randomized PCA was considered but ignored since it does not exploit the output label data to minimize feature space. LDA was also considered before being discarded. To this effect, feature selection step is performed using random forests with regression. Linear regression models such as Lasso and ElasticNet were considered before we discarded them for poor performance as measured by the coefficient of determination [4]. Recursive feature elimination techniques (with and without cross validation) were considered as well but were abandoned due to poor coefficient of determination values. These approaches involve removing one feature at a time and determining model performance on the remaining feature set at each step. The feature that impacts the model the best (i.e., model performance suffers upon that

[4]Coefficient of determination values were close to 0, far from being optimal.

feature removal) is retained for future use. Random forests are used to solve classification and regression problems. The functioning of random forests is described in detail in [5]. Random forests is an ensemble machine learning technique which uses several trees (estimators) to predict the outcome of test data. A tree is constructed from sample data selected from the training data. At each terminal node of the tree, m features are selected out of the total features and a best feature is identified for the tree to be split at. The tree is then split into child nodes. This is repeated until the selected sample size from the training data is the least. By using several trees and averaging the predictions, the variance across the trees is reduced. Mean squared error (MSE) is used to determine the best number of estimators (number of decision trees) used in the random forests algorithm. Different number of estimators such as 10 to 100 in steps of 5 are used in creating different random forest models. MSE is determined for each estimator and the average of the number of estimators is used as the MSE value for that specific estimators' number. The variation in MSE is before and after feature reduction is illustrated in Figure 25 (a) for one single case of network size 400 nodes at 90% loss and can be noticed that before feature reduction MSE is lowest when the number of estimators used in the random forest estimator is 70, and for after feature reduction MSE is lowest when the number of estimator is 50. The estimator for which MSE is the least is selected for calculating feature importances. In Figure 25 (b) for feed-forward loop connected motifs model for network size 400 nodes at 90% loss, we can notice that before feature reduction MSE is lowest when the number of estimators used in the random forest estimator is 95, and for after feature reduction MSE is lowest when the number of estimator is 25. Detailed explanation for the feature importances is left out due to space considerations [5]. At every run, feature importances, coefficient of determination and corresponding mean squared error change due to the randomization in the algorithm. To negate this, we

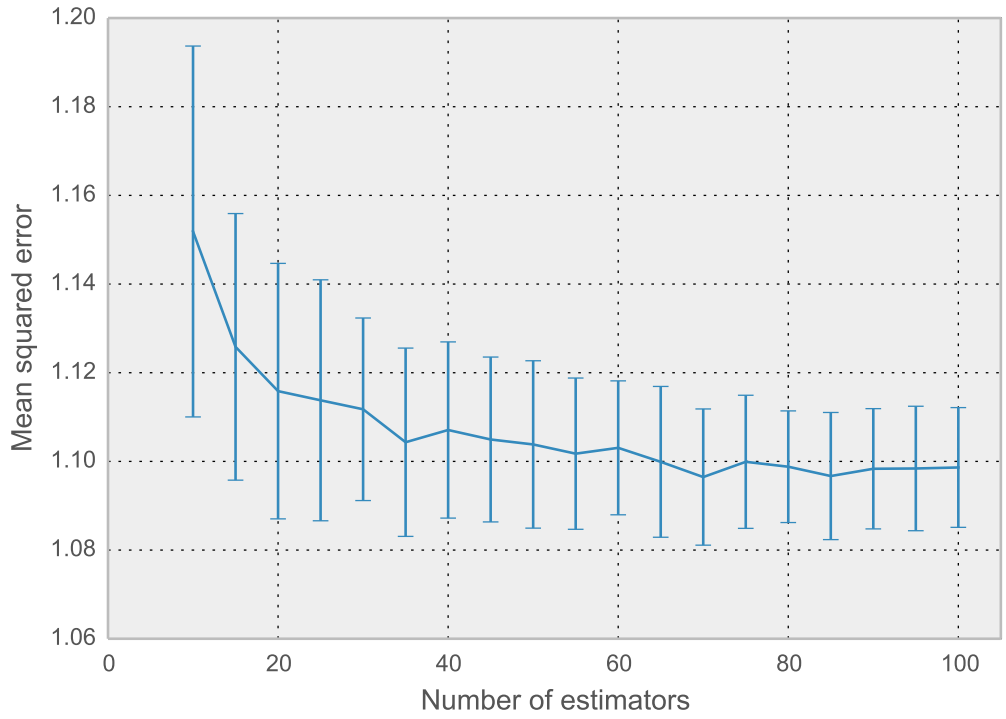


Fig. 25.: Mean square errors (MSE) at different estimators for 400 network size at 90% loss. Measured for the model with 38 features. Errorbars capture the variation of MSE across hundred test runs. Note that the Y-axis does not start at 0. *Lower MSE is better.*

execute the entire process for hundred runs and take the average of the respective values.

Our experiments reveal that the importance of features depends heavily on network size and loss it entails over time. Average of feature importances is used as a heuristic to select subset of thirty eight features. All the features with feature importance values greater than and equal to the average feature importance value are selected to model the final regressor for prediction.

5.3.3 Feature reduction

Coefficient of determination (COD) is used as a metric to measure a model's performance. For each of the thirty five random forest models, COD is determined before and after feature reduction. Each random forest regression model uses X number of estimators as shown in Figure 25. *Feature importance* of all the features is determined by averaging the total reduction in node impurity [5] across X estimators. We then create a random regressor to predict outcomes based on the model with reduced feature set which is tested using test data set.

COD measures the performance of predicted values by the model when compared to the real values. Good regressors will have a COD value close to 1 and the bad ones will have a COD close to 0. As evident from Figure 26, performing feature selection to reduce the feature set as explained in Section 5.3.2 does not improve the model accuracy. The majority of the models with all 38 features perform better than the models with a reduced feature set. The figure illustrating COD performance for models with 23 vertex-shared motif features is omitted as it follows similar trend.

Figure 27(a) presents the number of features selected by the feature selection process from all thirty eight features. It can be observed that the maximum number of features selected as important are 16 for the network size 200 at 50% loss and the least number of features that are selected as important are 3 for network 400 at loss 90%. At high loss (90%), few features (≤ 6) are responsible for network robustness.

Figure 27(b) shows important features selected from vertex-shared motifs for all network sizes at different loss scenarios. The number of significant features varies between 3 and 9. At high loss (90%), few features (≤ 5) are responsible for network robustness.

[5] as used in scikit-learn toolkit

5.3.4 Feature value correlation with robustness

In order to test the hypothesis if high feature values directly correlate with high robustness, we perform the following tasks. These tasks are executed at a network level. That is, significant features are identified for all models at different loss types for a given network size.

1. First, we identify the top five features using random forest regression (feature importance as a metric).

2. We then calculate the number of times each of the features occurs in the top five ranks at different loss scenarios.

3. Further, we determine the mean of each feature for a given model and identify the top five features with highest mean.

4. We then compare these features with the features obtained in second step.

As a result, we found no correlation (direct or inverse) between feature value and its importance. Among the models with all 38 features, gene percentage, direct FFL edge abundance, FFL indirect edges that participate in successful packet transmission to sink nodes, and the occurrences of direct edges in feed-forward loop motif (IDs 6, 8, 11, 12 respectively in Figure 28) are strong indicators of robustness. Apart from these features, network density, average shortest path, average degree centrality, and percentage of transcription factors (IDs 0, 1, 2, 3) also correlate to robustness relatively well. It is important to note that certain features make their impact distinctively in specific network sizes or at specific loss scenarios. This can be attributed to the fact that these specific features might be expressed more during the network extraction step (Section 5.2.2). The distribution of feature importances (with feature

IDs mentioned earlier) determined using random forest regression is shown in Figure 29. Each feature contains of hundred test runs to normalize the variations in feature importances due to randomization in regression algorithm. Outliers in the dataset are points that do not occur in the range of top and bottom whiskers and are identified by +.

5.4 Vertex-shared motifs

The importance of features as determined in Section 5.3.2 is charted in Figure 28. Heat maps are generated for all the networks at losses 10%, 20%, 35%, 50%, 60%, 75%, and 90%. Figure 28(a) represents one such case at 60% for model created with all 38 features. At one glance, it can be observed that features with IDs 1 to 13 and 28 stand out in all the networks. These features are *average shortest path*, *source to sink edge percentage*, *abundance of indirect FFL paths*, *percentage of direct FFL edges*, *percentage of indirect FFL edges*, *abundance of direct FFL edge occurrences*, and *abundance of indirect FFL path occurrences* respectively [6]. RH-7 (from Table 5) ranks as a significant feature in all network sizes and other connected motifs such as BW-4, RH-2 and BT-2 only stand out once.

Extending the hypothesis test described in Section 5.3.4 to models with only vertex-shared motifs (23 features), we found no correlation between feature value and its importance. Here, BW-1, BW-2, BW-4 and RH-7 (Refer to Table 4 and Table 5) are the strongly expressed features with robustness in all network sizes at different loss models. The results indicate that controlling the presence of these features can significantly impact biological network robustness. These features can also assist in creating superior bio-inspired networks where signal transduction is influenced by

[6]These features are described in our earlier work [24]

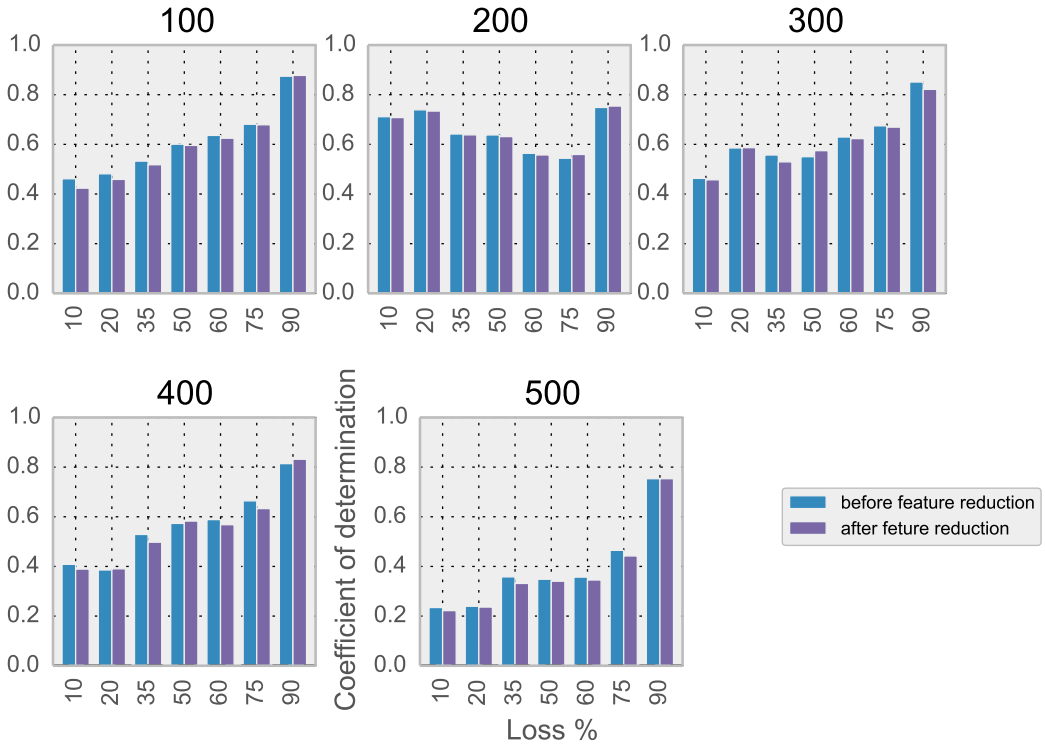
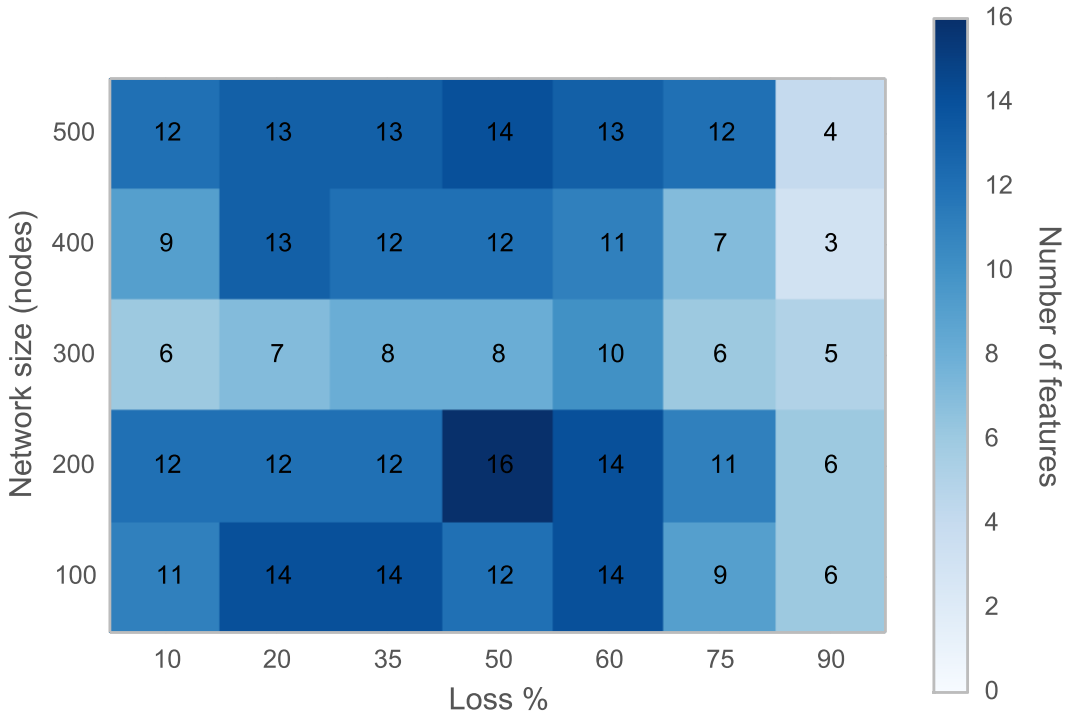


Fig. 26.: Coefficient of determination (COD) for regressors - different network sizes for 38 features model. Each data point represents an average value across 100 runs. *Higher COD is better* [61].

selective features such as the ones derived from FFL motif and the network itself can be adaptive by activating different regions at different periods of time to conserve energy.

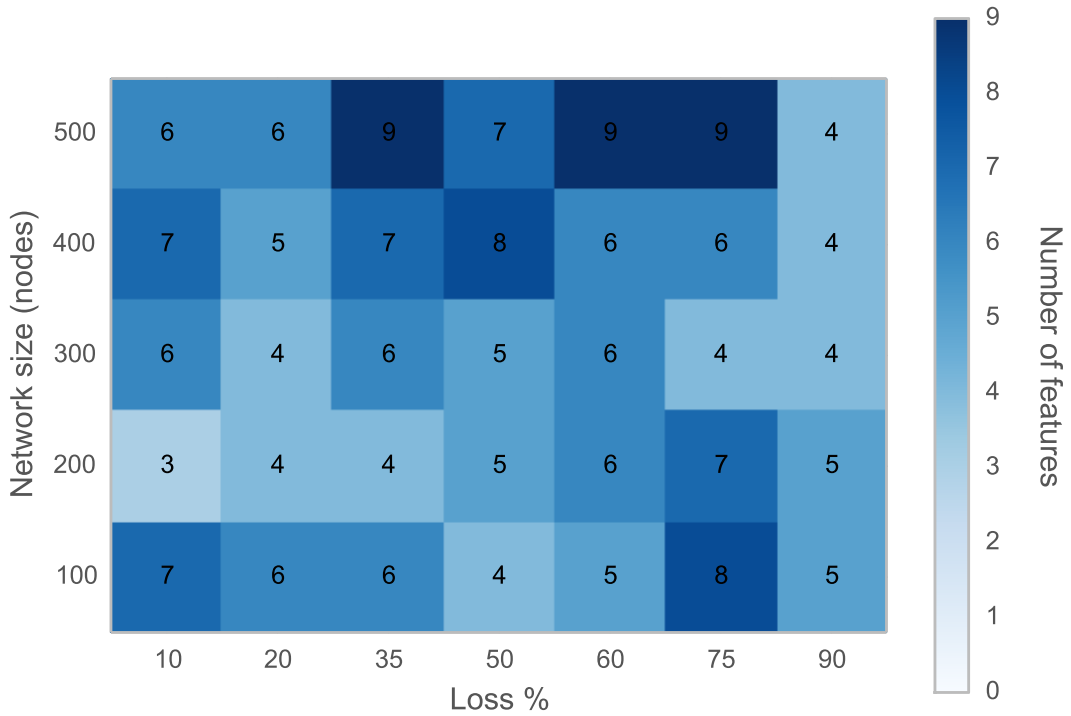
Figure 28(b) represents heat map of model created with twenty three features of feed-foreard loop connected motifs at loss 60%. This heatmap shows that Feature IDs BW-1, BW-2, BW-4, BW-6 (in two instances), RH-13 and BT-2 mark their presence in all the networks, but RH-7 ranks out as very important feature in all networks.



(a)

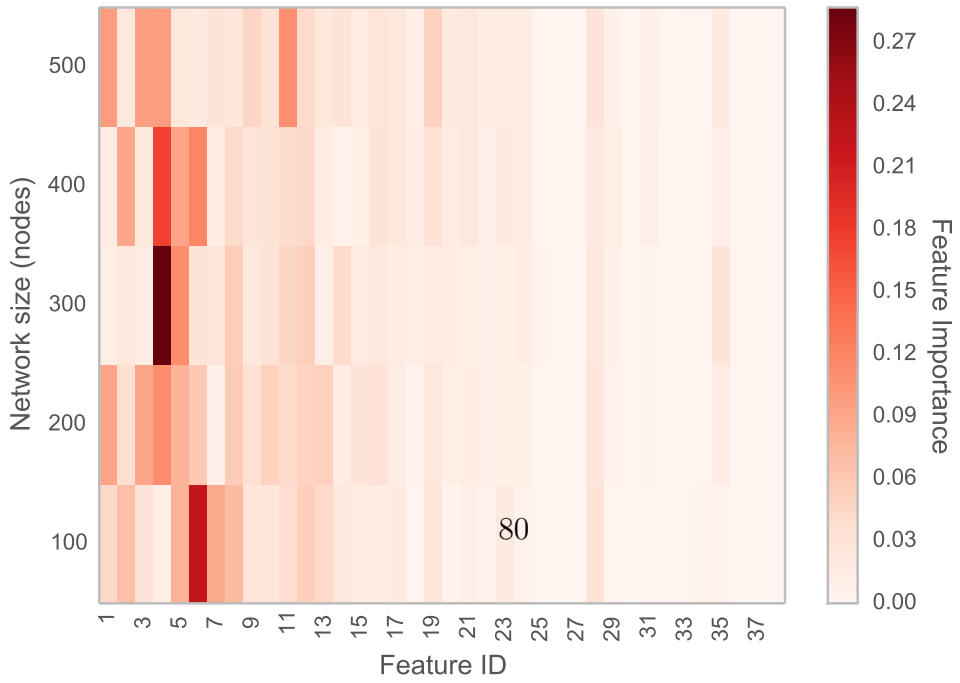
5.5 Discussion

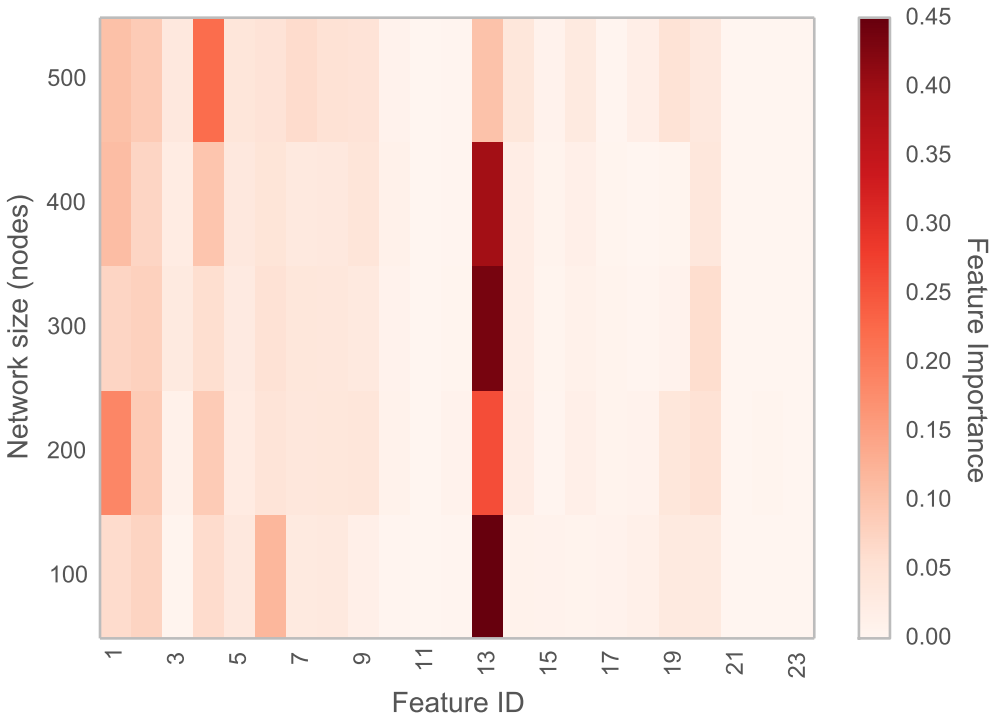
There is no one model that fits all data. We will extend the experiments to larger sized networks for *E. coli* transcriptional networks until maximum possible size is reached (i.e. number of nodes in *E. coli*) to explore if the trends in feature significance holds true. Further, we intend to extend the experiments to *Saccharomyces cerevisiae*. Our earlier experiments [24] revealed that feature significance varies from one model organism to the other and across network size and perturbation conditions. The higher ranking of FFL-derived features (IDs 7, 11, 12 in Figure 28) reveals the significance of motif derived features across different network sizes. Topological features such as network density, average shortest path remain important across all network sizes and under different loss conditions. The significance of vertex-shared motifs is relevant at high loss making them useful for constructing robust smart net-



(b)

Fig. 27.: (a) Selected features (out of total 38) for every model at a given network size and loss model as described in Section 5.3.2. (b) Selected features (out of 23) feed-forward loop connected motifs. Each data point represents an average value across 100 runs. Criteria: select features that have higher than average feature importance using random forest regression [61].





(b)

Fig. 28.: (a) Feature significance in all the networks at 60% loss for model with all 38 features. (b) Feature significance of connected feed-forward loop motifs in all the networks at loss 60%. *The darker the color the higher the feature significance.* Additionally, numbers are included to indicate feature rank. Each data point represents an average value across 100 runs. *Higher the feature importance, better is the feature* [61].

works capable of surviving lossy conditions. New research has indicated the evolution of bow-tie motif under distinct conditions such as a limitation on number of edges in a network [13] and its potential role in maintaining biological network robustness [62].

This is an interesting proposition for designing engineered systems that exploit the principles seemingly intrinsic to the design of biological network topologies. The implications of specialized engineered systems cannot be ignored in the areas of disaster relief coordination.

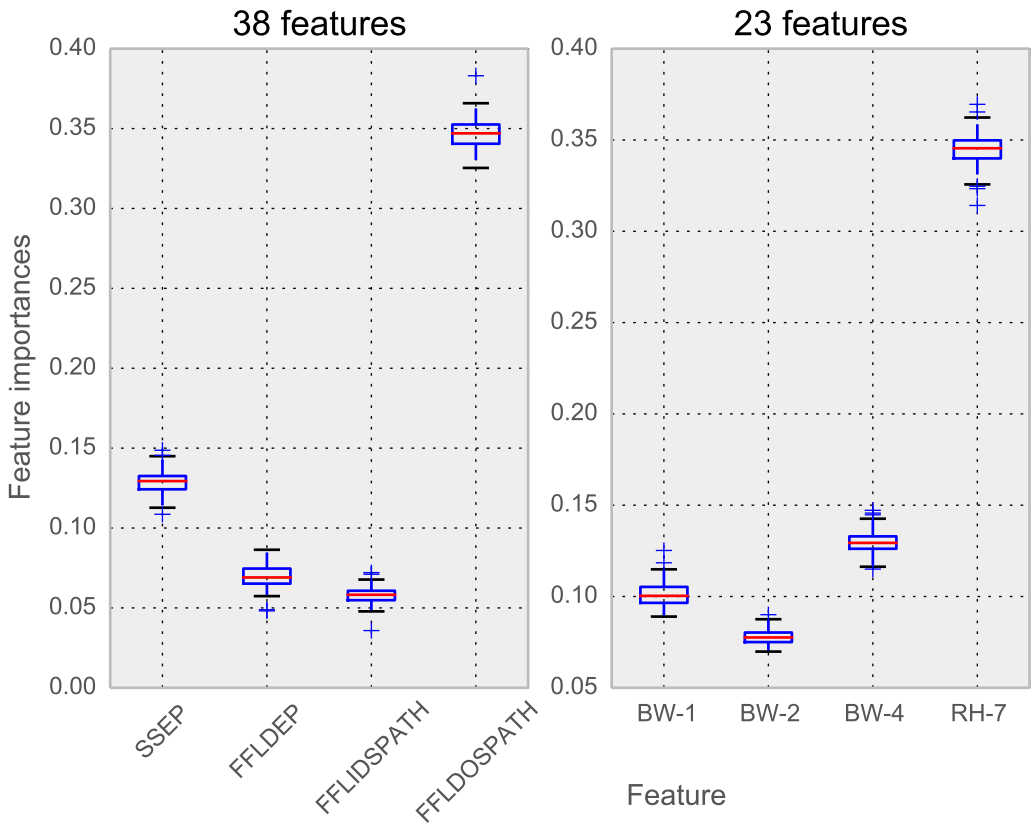


Fig. 29.: Feature importances as determined by random forest regression for models with 38 features and 23 features respectively for network size 100 at 75% loss. While the feature source to sink edge percentage is termed as SSEP, the percentage of FFL direct edges is termed FFLDEP. The features 10, 11 as explained in Section 5.2.4 are FFLIDSPATH and FFLDOSPATH. Refer to Table 4 and Table 5 for definitions of BW-1, BW-2, BW-4, and RH-7. *Higher feature importance is better.*

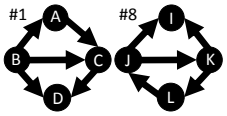
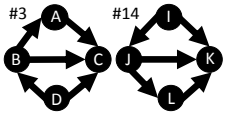
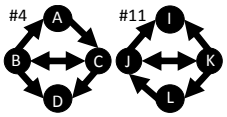
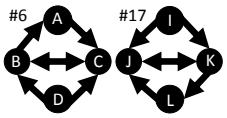
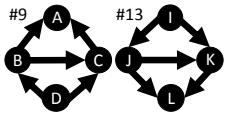
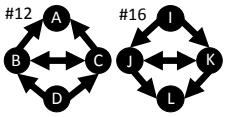
Table 4.: Abundance of bow-tie and bi-triangle motifs in *E. coli* transcriptional network [61].

Pattern ID	**Symbol**	**Abundance**
BW-1		139827
BW-2		110505
BW-3		730
BW-4		24032
BW-5		1412
BW-6		1393
BT-1		17
BT-2		439
BT-3		4
BT-4		140
BT-5		3

Table 5.: Abundance of rhombus motifs in *E. coli* transcriptional network [61].

Name	**Symbol**	**Abundance**
RH-1		623
RH-2		553
RH-3		788
RH-4		93
RH-5		7
RH-6		9
RH-7		69299
RH-8		516
RH-9		58364
RH-10		200
RH-11		656
RH-12		30

Table 6.: Isomorphic rhombus motifs in *E. coli* transcriptional network [61].

Name	**Symbol**
RH-1/RH-8	
RH-3/RH-14	
RH-4/RH-11	
RH-6/RH-17	
RH-9/RH-13	
RH-12/RH-16	

CHAPTER 6

ROLE OF FFLS IN SIGNAL TRANSDUCTION

Following the relatively insignificant contribution of vertex-shared motifs to network robustness, we reduced the number of features to 15 and studied them further. We followed the similar procedure explained in Chapter 5 to estimate network characteristics that contribute to robustness using random forest regression. Table 7 presents the mean and standard deviation values for the 1000 samples used for 100 network size. Additonally, Tables 8, 9, 10, 11 in Chapter 8 show similar statistics for network sizes 200, 300, 400 and 500. Figures 30, 31, 32, 33, 34 plot the feature importance of all the features. Each of the feature value mentioned in the heat map cell is an average across 100 runs. Figures 35 and 36 illustrate the distribution of feature importances across 100 runs for 500 network size at two perturbation levels (20% and 50%). Illustrations for feature importance distribution for other network sizes and at different perturbation levels are provided in the 8. Figure 38 shows the coefficient of determination distribution of the machine learning models before feature reduction as explained in earlier chapter. Chapter Appendix also illustrates another similar distribution plot for coefficient of determination before feature reduction.

It can be noticed from all the figures that the feature **FFLIDEP** emerges as a strong feature at higher loss. Features like *Density*, *TFP* and *GP* can be observed to be more important than others in multiple instances. We can recollect that *FFLIDEP* feature is the percentage of indirect FFL edges that are present in the network compared to the total edges. Our primary hypothesis is to test if the reduction in feature importance of *FFLDEP* at higher loss leads to higher importance of *FFLIDEP*. We

explicitly observe this in Figure 37 below. We realized that our hypothesis is true for all network sizes except 100. Following this, we intended to study the entire *E. coli* regulatory network for these patterns and introduced a new way to understand FFL distribution as described below.

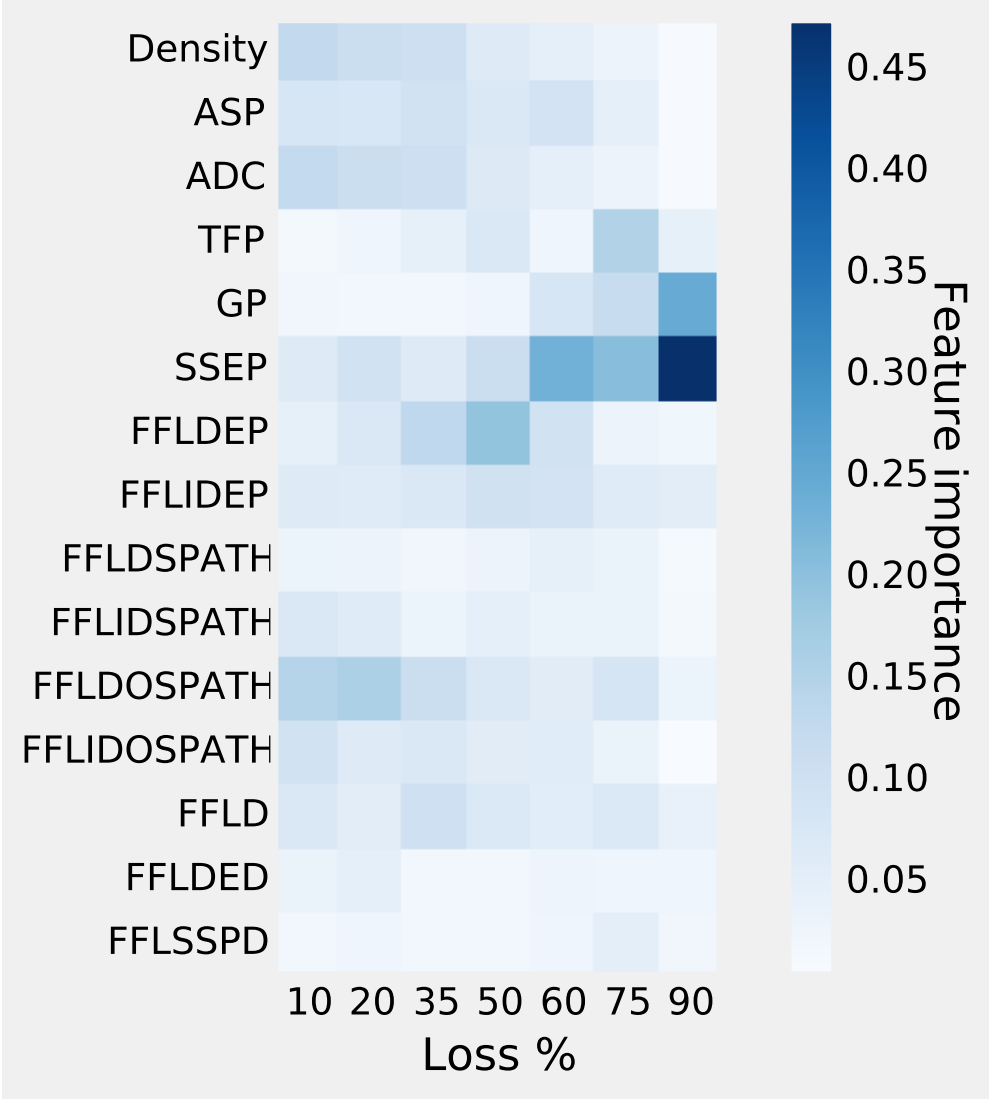


Fig. 30.: Feature importance - 100 network size. SSEP feature stands out with increase in noise. Each of the feature value mentioned in the heat map cell is an average across 100 runs.

Figure 39 is used as a reference to explain the following concepts. In order to perform this, we look at FFLs that follow, what we term, shortest path switch. Shortest path switch is defined as follows: shortest path from node A to node C is

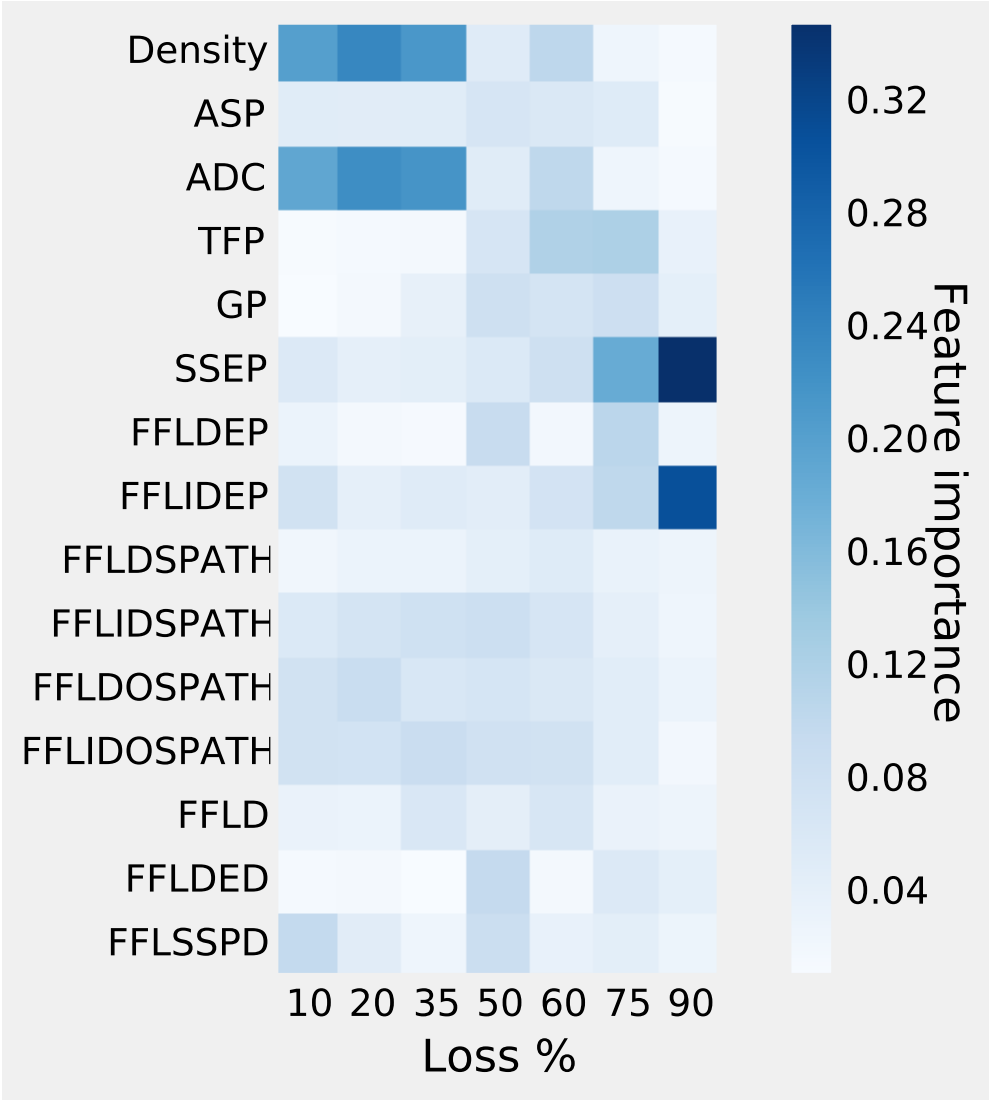


Fig. 31.: Feature importance - 200 network size. While network density and average degree centrality (ADC) stand out upto loss 35%; SSEP, FFLIDEP features stand out with increase in noise. Each of the feature value mentioned in the heat map cell is an average across 100 runs.

always via the direct edge. However, under heavy noise the direct edge of FFL will potentially be destroyed making the information flow from node A to node C occur via the indirect path (via node B). Here we identify all FFLs that switch the shortest

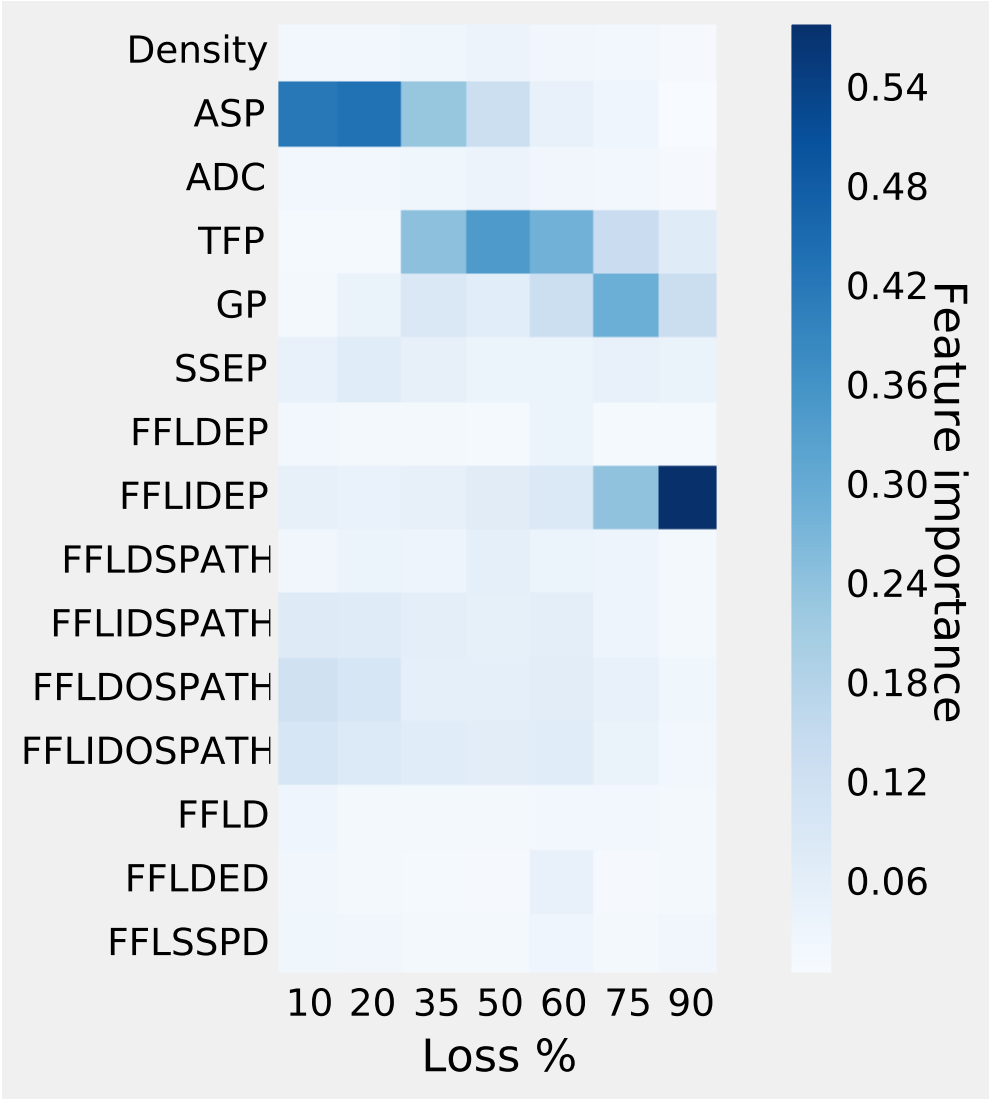


Fig. 32.: Feature importance - 300 network size. The percentage of transcription factor and gene nodes are important at higher noise. FFLIDEP feature stands out with increase in noise. Each of the feature value mentioned in the heat map cell is an average across 100 runs.

path from direct to indirect FFL edge.

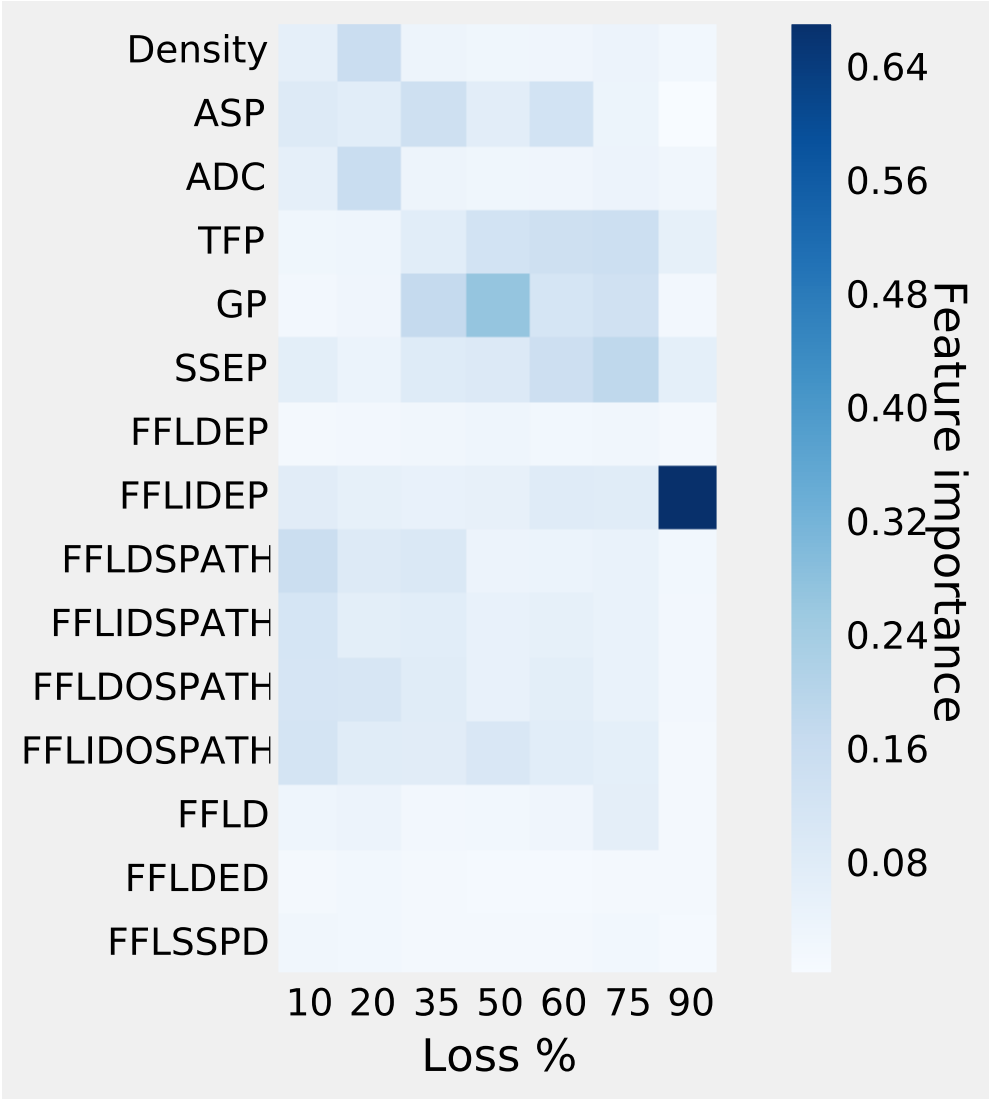


Fig. 33.: Feature importance - 400 network size. FFLIDEP feature stands out with increase in noise. Each of the feature value mentioned in the heat map cell is an average across 100 runs.

6.1 Signal transduction

We group FFLs into two categories: canonical and embedded. FFLs with no additional edges among the nodes are considered to be canonical. FFLs with addi-

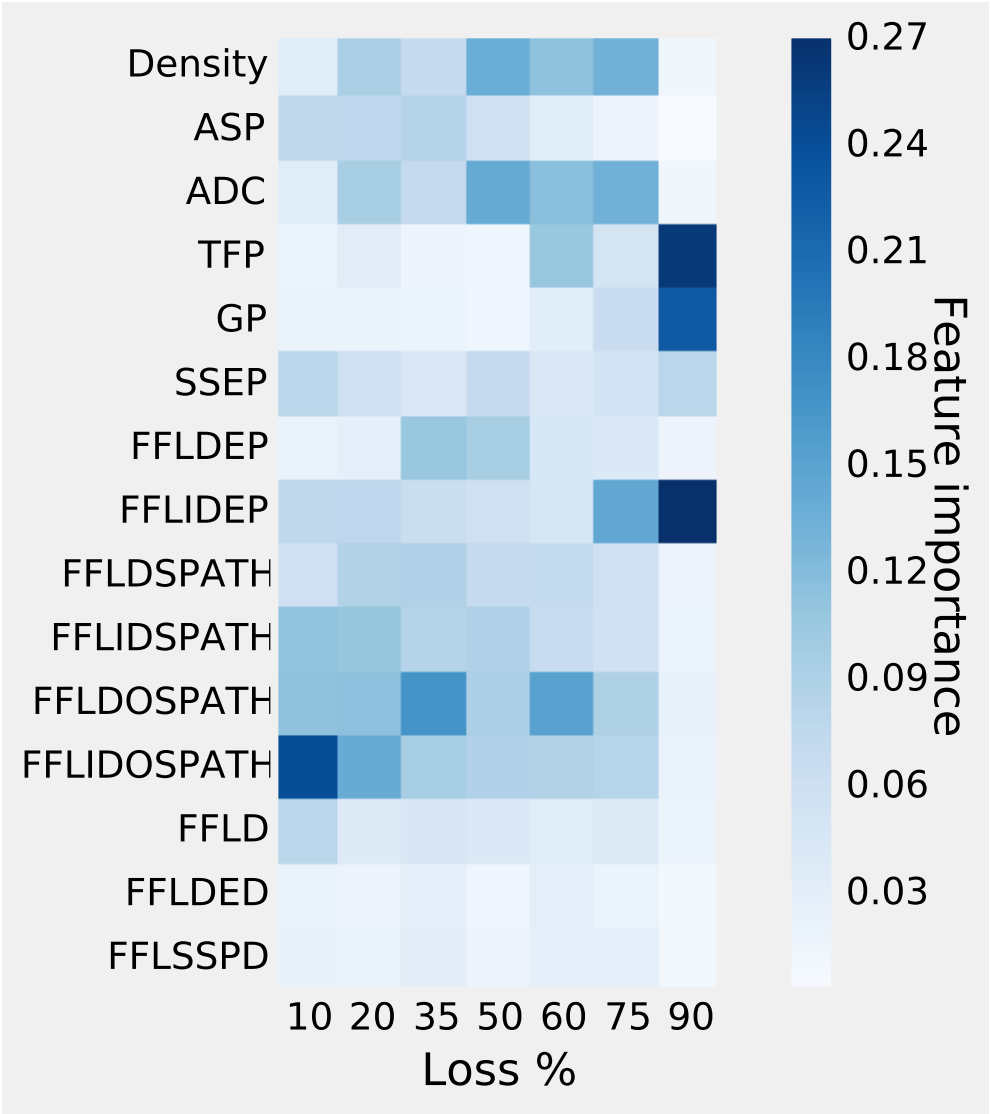


Fig. 34.: Feature importance - 500 network size. FFLIDEP along with other FFL-derived features stand out with increase in noise. Each of the feature value mentioned in the heat map cell is an average across 100 runs.

tional edges among the nodes are considered to be embedded. This is illustrated in Figure 39. Further, we group each of these FFL categories into peripheral and non-peripheral FFLs. Peripheral FFLs are FFLs in which the node being transcribed has no out degree. Non-peripheral FFLs are FFLs in which the nodes being transcribed have non-zero out degree.

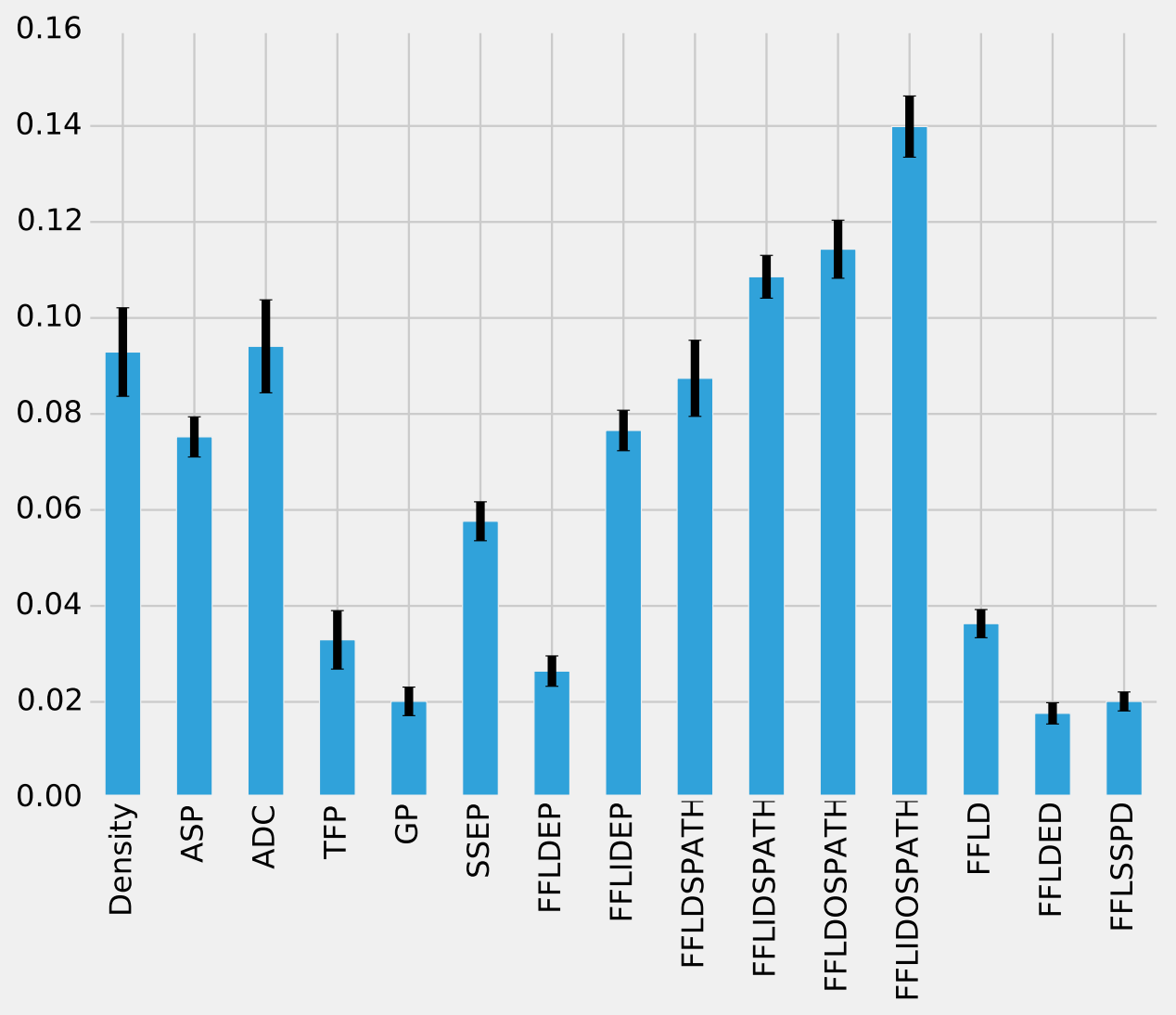


Fig. 35.: Distribution of scaled feature importance values - 500 network size at 20% perturbation level. Each feature is measured across 100 runs.

Our interest to study FFLDEP and FFLIDEP in particular is due to the fact that these two motif-derived features effectively capture the FFL path switch from direct to indirect for information transmission. Our idea to identify FFLs that are central to the *E. coli* transcriptional regulatory network led to the study of distribution of peripheral and non-peripheral categories of canonical and embedded FFLs. First,

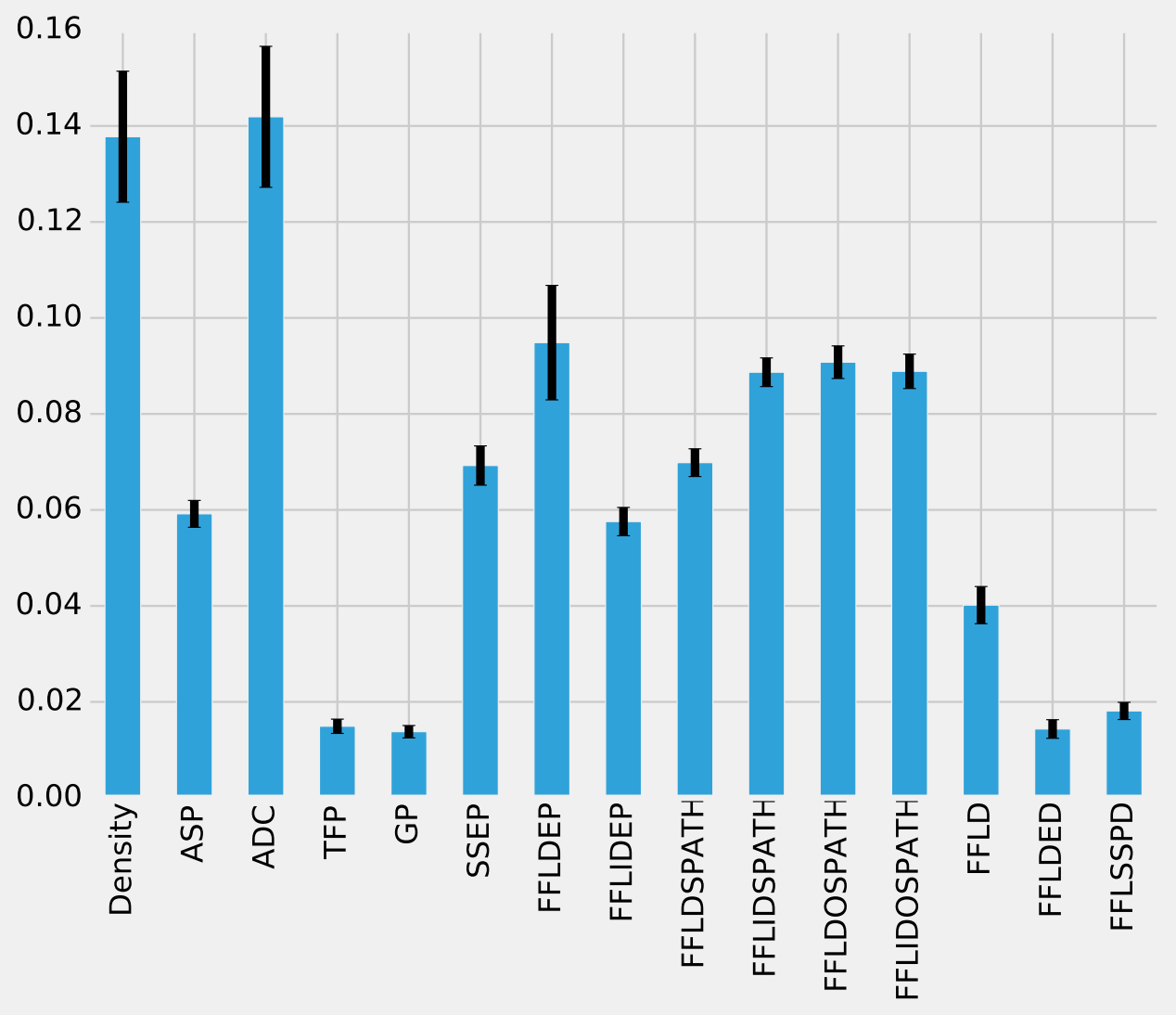


Fig. 36.: Distribution of scaled feature importance values - 500 network size at 50% perturbation level. Each feature is measured across 100 runs.

we identified all canonical and embedded FFLs. A majority of FFLs (64.5% and 80.5% respectively) switched paths on edge deletion in canonical and embedded FFLs. Figure 40 presents the detailed distribution of the physical location of FFLs within *E.coli* network. This reveals that only a small number of FFLs (6 canonical non-peripheral and 26 embedded non-peripheral FFLs) participate in signal transmissions within this network. Effectively, only the gene nodes in these filtered FFLs have outgoing edges enabling them to participate in signal transmission. Controlling the

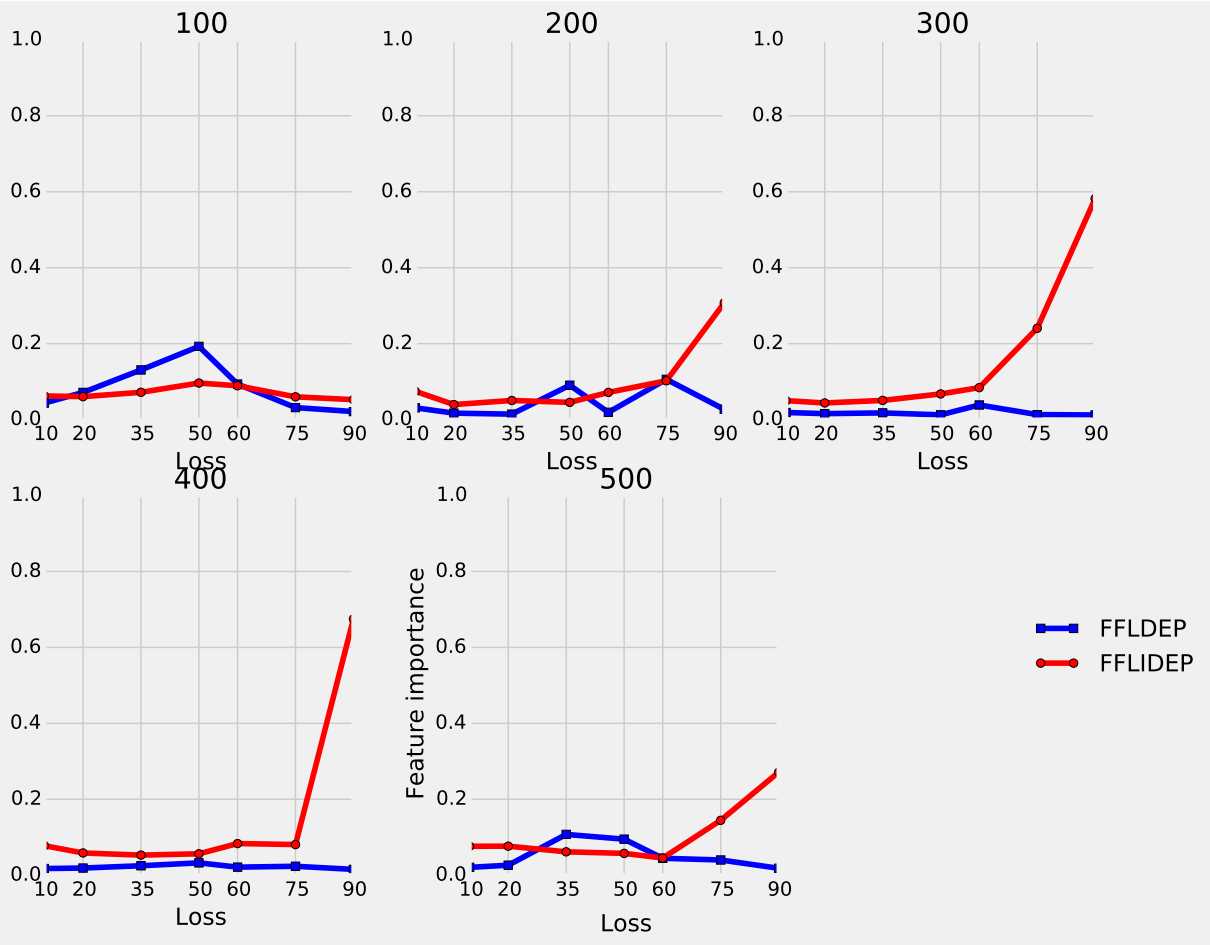
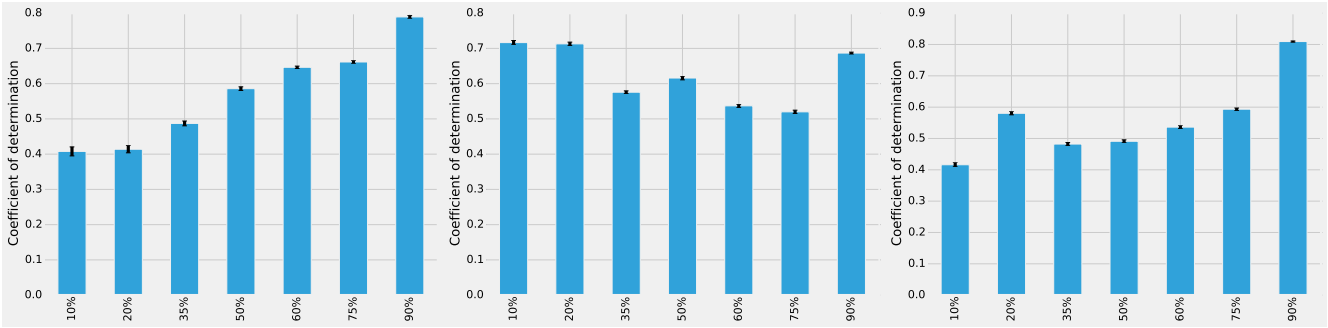
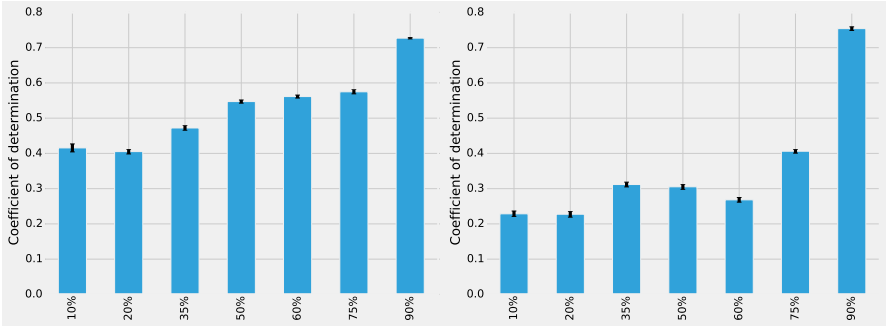


Fig. 37.: Relative feature importance of FFLDEP vs FFLIDEP - all network sizes at different perturbation levels. Each data point is an average of 100 runs.



(a) 100 (b) 200 (c) 300



(d) 400 (e) 500

Fig. 38.: Distribution of coefficient of determination after feature reduction - all network sizes at different perturbation levels. Each feature is measured across 100 runs.

nodes in these filtered FFLs can establish critical patterns prevalent in the regulatory network.

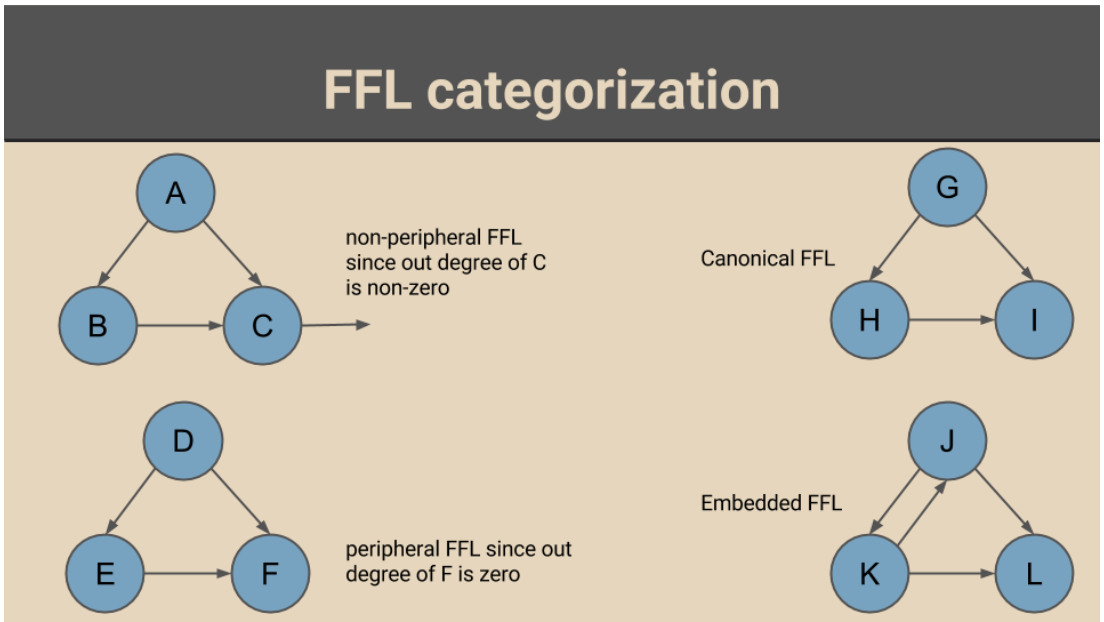


Fig. 39.: Categorization of peripheral and non-peripheral FFLs into canonical and embedded FFLs. An FFL is peripheral if the out degree of gene node (sink) is zero and is non-peripheral otherwise.

Type of FFL	# of FFLs (% of corresponding FFLs)	# of FFLs with path switch (%)
Canonical	956 (51.4%)	617 (64.5%)
Embedded	904 (48.6%)	728 (80.5%)
Canonical non-peripheral	24 (2.5%)	6 (25%)
Embedded non-peripheral	76 (8.4%)	26 (34%)
Canonical peripheral	932 (97.5%)	611 (65.5%)
Embedded peripheral	828 (91.6%)	702 (84.78%)

Fig. 40.: Distribution of peripheral and non-peripheral FFLs into canonical and embedded FFLs in E. coli transcriptional regulatory network.

Table 7.: Mean and standard deviation (STD) of features - 100 network size at under different perturbation levels.

Feature	Mean at 10%	STD at 10%	Mean at 20%	STD at 20%	Mean at 35%	STD at 35%	Mean at 50%	STD at 50%	Mean at 60%	STD at 60%	Mean at 75%	STD at 75%	Mean at 90%	STD at 90%
Density	0.003743207	3.84E-07	0.0037469121	3.97E-07	0.003744793	3.96E-07	0.0037338029	3.84E-07	0.0037251903	3.60E-07	0.0037113202	2.30E-07	0.0041664741	1.06E-07
Average shortest path (ASP)	0.0096331775	7.99E-05	0.0097146137	8.01E-05	0.0096803428	8.05E-05	0.0089164783	8.22E-05	0.0073056009	7.19E-05	0.0035478256	5.37E-05	0.0001272936	1.42E-06
Average degree centrality (ADC)	0.0074686415	1.54E-06	0.0074938241	1.59E-06	0.0074589586	1.58E-06	0.0074676058	1.53E-06	0.0074503806	1.44E-06	0.0074226405	9.20E-07	0.0083329481	4.24E-07
Transcription Factors Percentage (TFP)	0.0973524684	0.000209932	0.0973177471	0.0002091796	0.0973177471	0.0002091796	0.0973177471	0.0002091796	0.0973177471	0.0002091796	0.0973177471	0.0002091796	0.0973177471	0.0002091796
Genes Percentage (GP)	0.9026475316	0.000209932	0.9026822529	0.0002091796	0.9026822529	0.0002091796	0.9026822529	0.0002091796	0.9026822529	0.0002091796	0.9026822529	0.0002091796	0.9026822529	0.0002091796
Source to Sink Edges Percentage (SSEP)	0.9503615947	0.0001648742	0.9503998831	0.0001639272	0.9503998831	0.0001639272	0.9503998831	0.0001639272	0.9503998831	0.0001639272	0.9503998831	0.0001639272	0.9503998831	0.0001639272
FFL Direct Edges Percentage (FFLDEP)	0.2877973011	0.0086393445	0.2885331128	0.0088325548	0.2885331128	0.0088325548	0.2885331128	0.0088325548	0.2885331128	0.0088325548	0.2885331128	0.0088325548	0.2885331128	0.0088325548
FFL InDirect Edges Percentage (FFLIDEP)	0.3412775178	0.0198585863	0.3419949506	0.0198167909	0.3419949506	0.0198167909	0.3419949506	0.0198167909	0.3419949506	0.0198167909	0.3419949506	0.0198167909	0.3419949506	0.0198167909
Number of Direct FFL edges in Successful Paths (FFLDSPATH)	0.8791536923	0.0079157161	0.8812921059	0.0072428312	0.8766208988	0.0075665767	0.8531760284	0.0092332891	0.8237746906	0.0098632073	0.6975529022	0.0112493938	0.3612801397	0.0046528756
Number of InDirect FFL edges in Successful Paths (FFLIDSPATH)	0.3387563681	0.0024116637	0.4767824709	0.0040657509	0.5380006734	0.0060312976	0.49990674	0.0065882516	0.4157241562	0.0057982786	0.2216269354	0.0033465038	0.0388199192	0.0003861267
Number of Direct FFL edge Occurrences in Successful Paths (FFLDOSPATH)	0.2667275881	0.0064073993	0.2794575629	0.00618499	0.2957932208	0.0051712638	0.3156789618	0.0044493569	0.3270474498	0.0043811476	0.3337160542	0.0043408458	0.3254285622	0.0062987348
Number of InDirect FFL edge Occurrences in Successful Paths (FFLIDOSPATH)	0.0434447229	0.0019989235	0.0774370522	0.0050521145	0.1238624521	0.0103187942	0.1708778386	0.0140421093	0.2035004227	0.0172920544	0.2516273368	0.0212381296	0.3054840744	0.0316843643
FFL edge Density (FFLD)	0.4247097409	0.0119757995	0.4253153947	0.0121345015	0.4253153947	0.0121345015	0.4253153947	0.0121345015	0.4253153947	0.0121345015	0.4253153947	0.0121345015	0.4253153947	0.0121345015
FFL Direct Edge Density (FFLDED)	0.3499201757	0.012229098	0.3507249231	0.0125090863	0.3507249231	0.0125090863	0.3507249231	0.0125090863	0.3507249231	0.0125090863	0.3507249231	0.0125090863	0.3507249231	0.0125090863
FFL Source to Sink edge Percentage Density (FFLSSPD)	0.367414175	0.0126037947	0.3682240686	0.01288717	0.3682240686	0.01288717	0.3682240686	0.01288717	0.3682240686	0.01288717	0.3682240686	0.01288717	0.3682240686	0.01288717

CHAPTER 7

CONCLUSION

Understanding the governing principles of biological networks is considered to be the holy grail of systems biology. We extensively studied the structural patterns in the transcriptional regulatory network of model organism *Escherichia Coli*. To this effect, I proposed and developed in-silico models to capture robustness for biological networks by extending the concepts conventionally defined for wireless sensor networks. After establishing the simulation framework, I extensively studied the features responsible for network robustness. To achieve this, I used machine learning techniques such as support vector machines, random forest regression modeling. We then looked at the contribution of vertex-shared motifs towards biological robustness. Our results indicated that barring couple of attack scenarios, vertex-shared motif structures do not contribute to network robustness. Our experiments revealed that the features derived from feed-forward loop motifs contributed strongly to network robustness. We also observed the strong contribution of well studied topological characteristics namely network density, average shortest path. We also studied the distribution of feed-forward loop motifs within the regulatory network which revealed that only small number of such motifs participate in signal transmissions within the network. This work paves the way to create special engineered networks which can possess these highly expressed features abundantly ensuring *robust* network behavior even in attack scenarios.

CHAPTER 8

FUTURE WORK

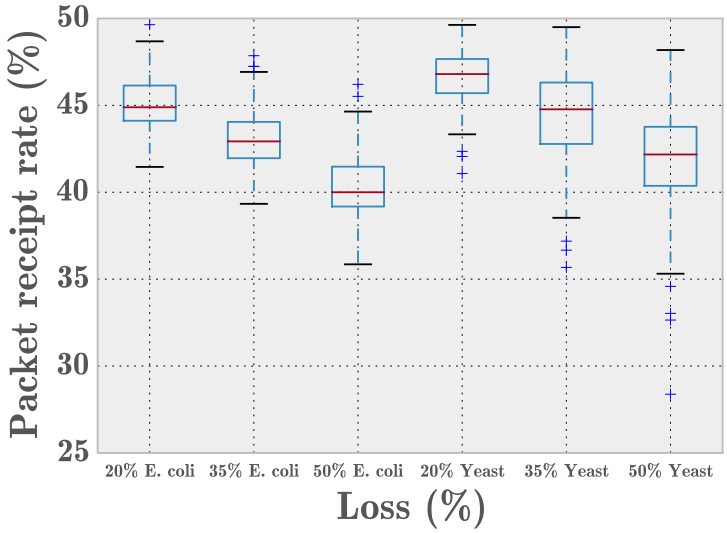
My work introduced a framework to quantify dynamic biological robustness using strategies from event driven network simulation, machine learning regression and classification, graph transformation and structural biological principles such as motifs. This is different from conventional attempts to capture system robustness in terms of only topological parameters such as network density and shortest path.

Biological system modeling is fragmented depending on the contextual problem. Currently there are software to model few aspects of biological information transmission such as brain and other organ-specific simulations. There are software such as Cytoscape [59] to visually explore biological interaction networks and identify function-specific modules and entities. While the contribution of Cytoscape-like software to understand biological structural topology is immense, we are far from developing a true simulator that can replicate biological systems. Considering gene regulatory networks as an example, the major obstacles to design a true model that replicates biological interactions is the number of unknown constraints within gene regulation. While network simulators such as NS-2 are not designed for studying biological interactions, they are a viable option until other models become available.

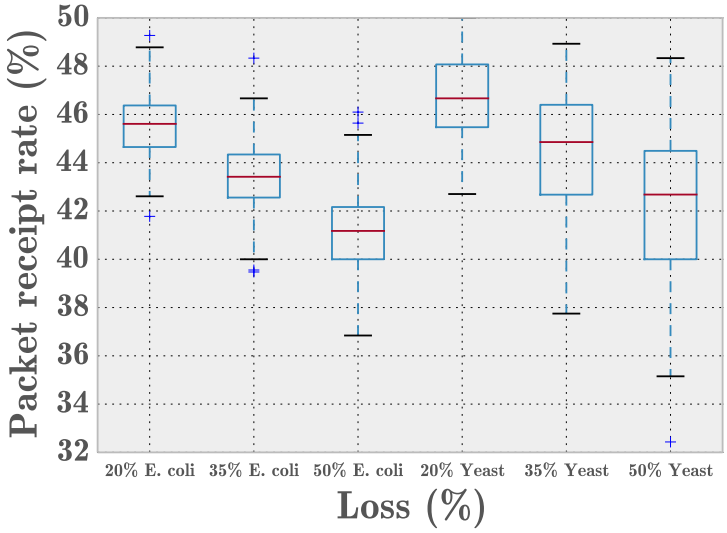
Building on this research, engineered networks can be created that are robust under lossy conditions. Algorithms that create these engineered systems must include the features identified as important in prior research. Researchers have proposed methods in the past to develop complex systems ensuring specific topological aspects-for instance, retaining overall degree distribution while growing networks in

Barabasi-Albert preferential attachment model [2]. Such models can be studied and improved complex network models that include feed-forward loop motif derived can be developed ensuring robust system behavior. Robust networks can also be engineered using the features our research identified as significant by machine learning techniques. These features include average network shortest path and feed forward loop motif. Each such structure contains two edge disjoint paths for a transcription factor node to regulate a gene node (in the context of a Gene regulatory network). In translation to the node context, the transcription factor node can send information to the gene node via two paths. This structure becomes prominent at high perturbation levels. Most canonical FFLs exhibit only a single shortest path to their sink that passes through FFL direct edge; under noise when this direct edge becomes unavailable, information transport switches to the indirect FFL path which alternately suggest that the shortest path to sink for this FFL has increased by 1-hop. While this work only explores the structural contribution of feed-forward loop motif, other motifs (bifan, for instance) are also shown to be promising in their contribution to biological network robustness. Recent research has highlighted the evolution of bow-tie motifs after identifying their role in biological signalling and in information processing [13]. Bow-tie motifs have a similar design to that of an hour glass. One central node has several incoming and outgoing edges. After processing the information from the incoming edges, it sends necessary information to outgoing edges. These motifs can introduce interesting dimension by adding new kind of nodes in bio-inspired network topologies. Combining the predictive power of machine learning algorithms with the time-series gene-transcription factor interaction data can throw new light on the time-dependent regulation complexities in regulatory networks.

APPENDIX

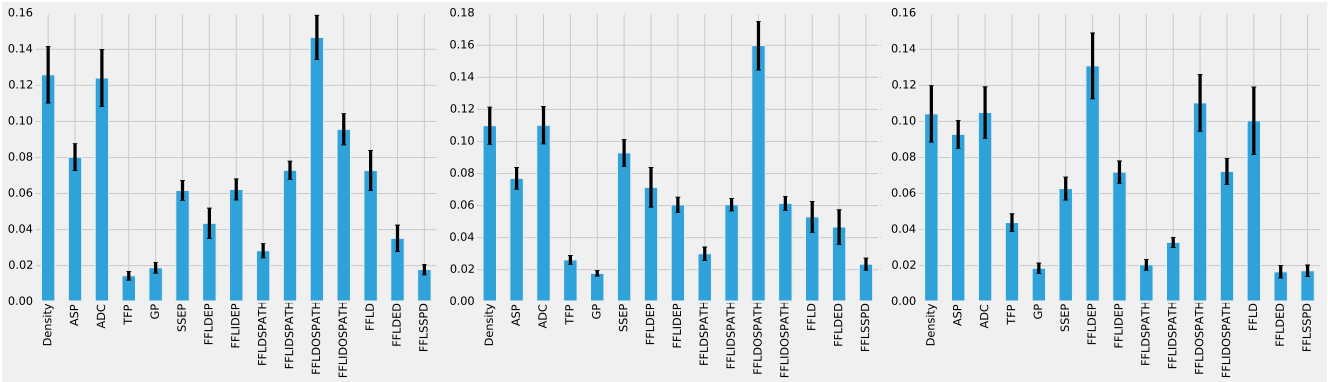


(a) $n = 400$ nodes.



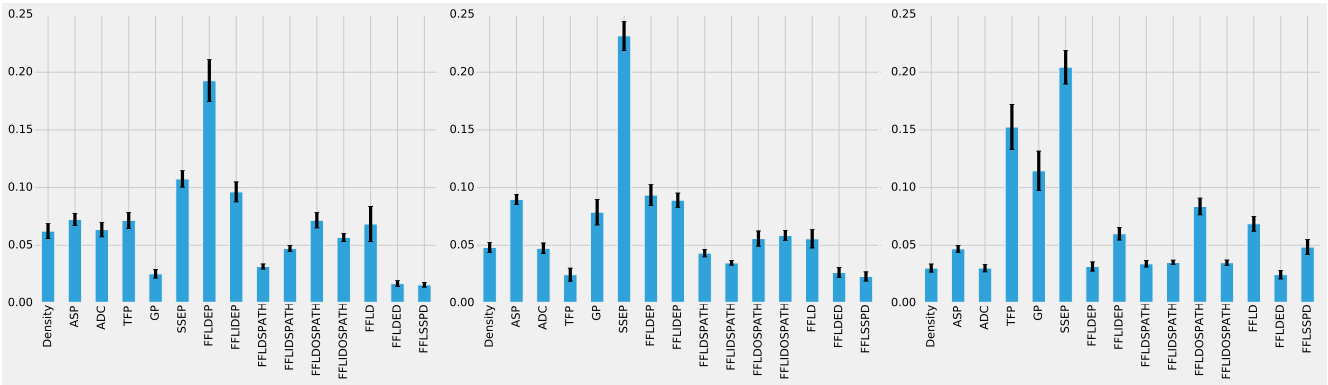
(b) $n = 500$ nodes.

Fig. 41.: Packet receipt rates (PRTs) for sampled transcriptional subnetworks of the bacterium *Escherichia coli* and *Saccharomyces cerevisiae* (labeled 'Yeast').



(a) 10%

(b) 20%



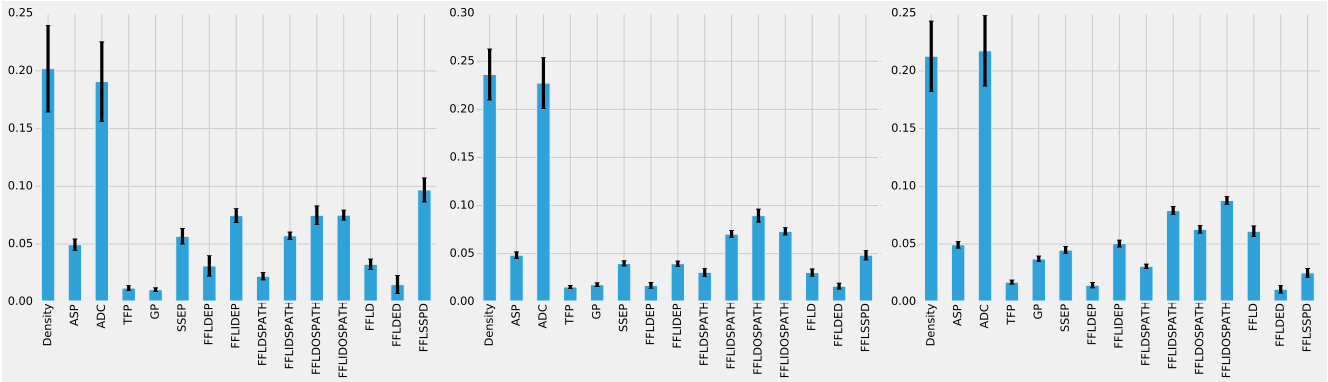
(c) 35%

(d) 50%

(e) 60%

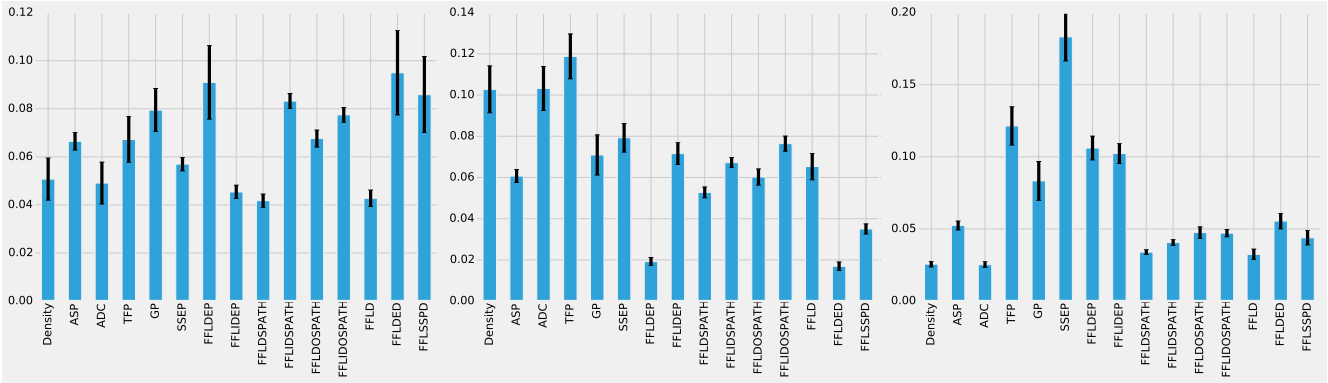
(f) 75%

Fig. 42.: Distribution of scaled feature importance values - 100 network size at different perturbation levels. Each feature is measured across 100 runs.



(a) 10%

(b) 20%



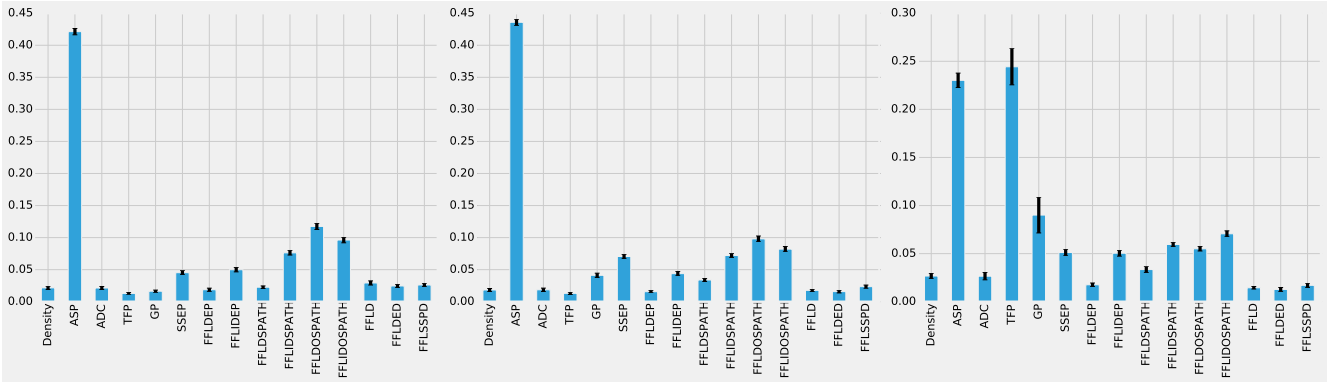
(c) 35%

(d) 50%

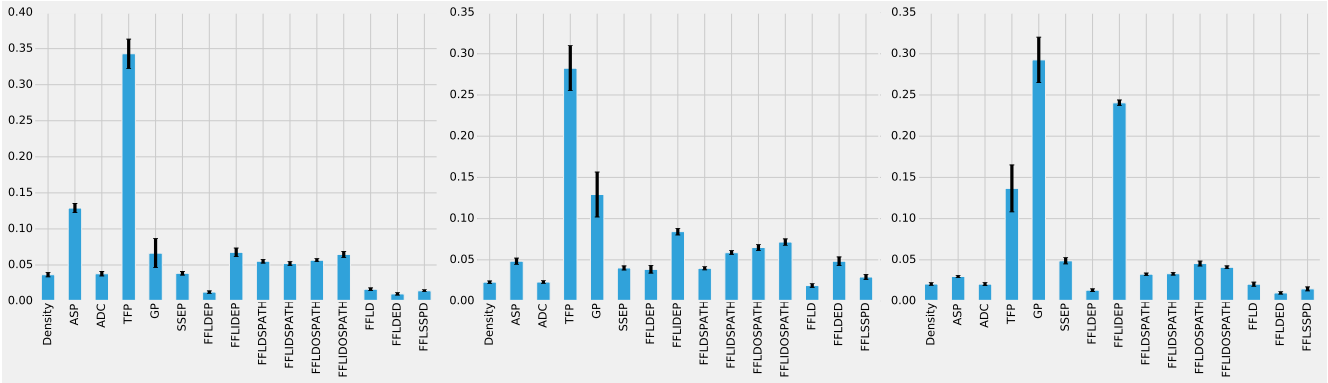
(e) 60%

(f) 75%

Fig. 43.: Distribution of scaled feature importance values - 200 network size at different perturbation levels. Each feature is measured across 100 runs.

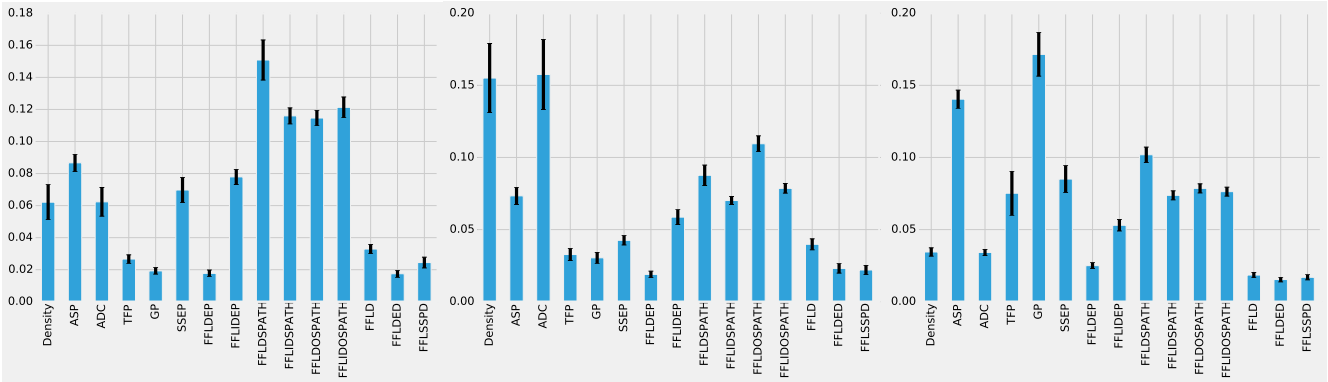


(a) 10% (b) 20% (c) 35%

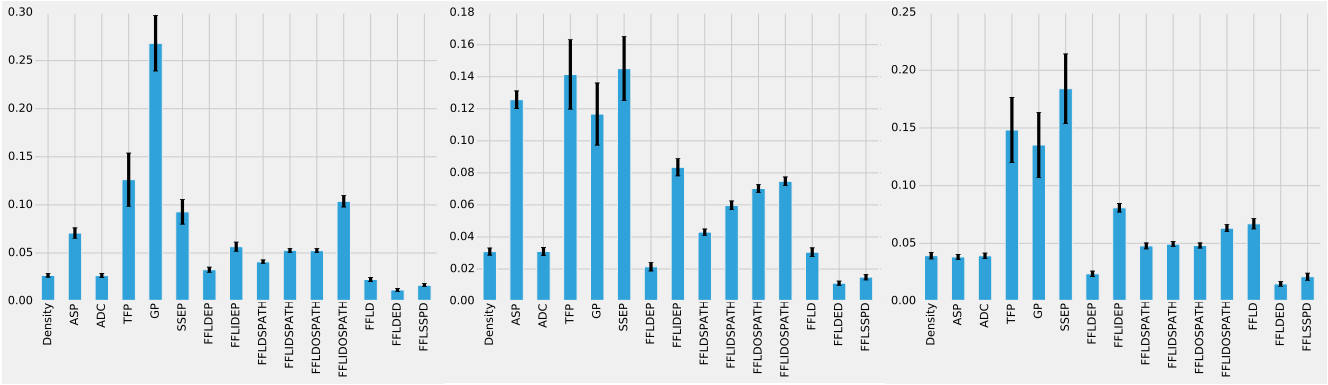


(d) 50% (e) 60% (f) 75%

Fig. 44.: Distribution of scaled feature importance values - 300 network size at different perturbation levels. Each feature is measured across 100 runs.

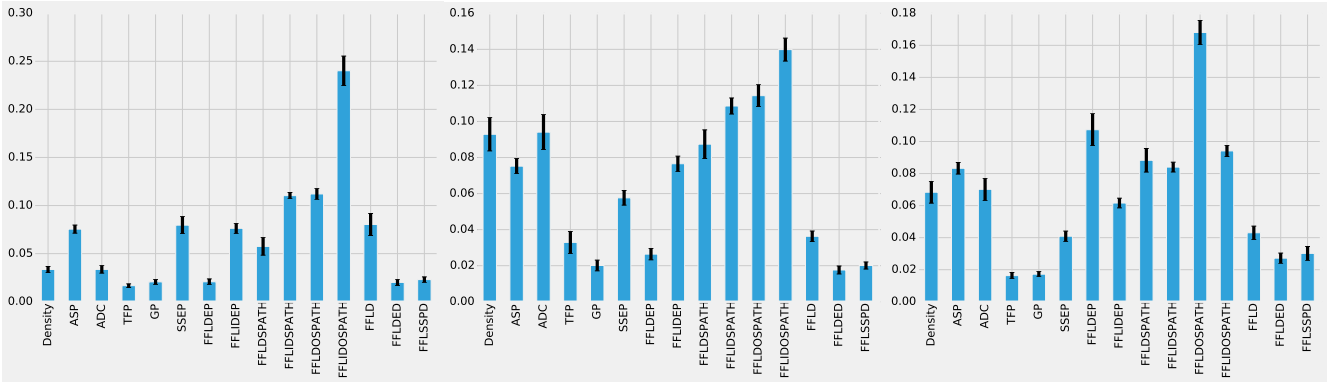


(a) 10% (b) 20% (c) 35%

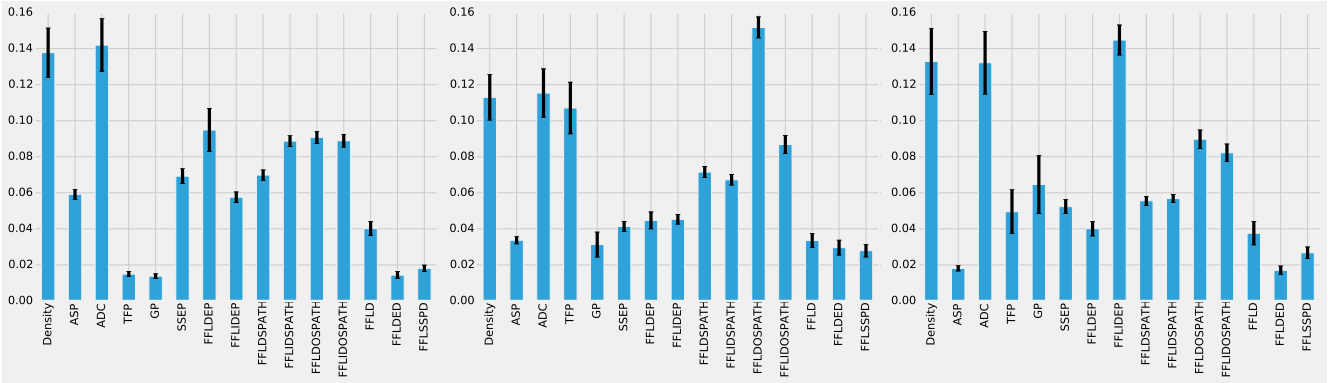


(d) 50% (e) 60% (f) 75%

Fig. 45.: Distribution of scaled feature importance values - 400 network size at different perturbation levels. Each feature is measured across 100 runs.

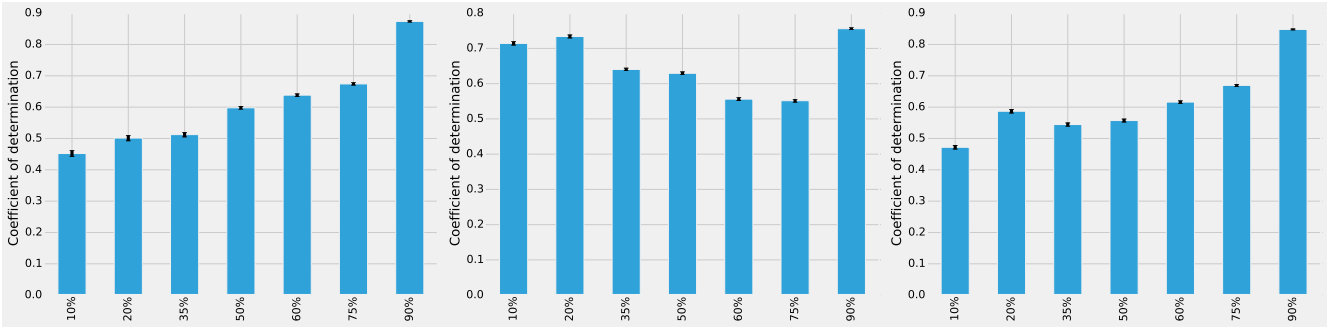


(a) 10% (b) 20% (c) 35%

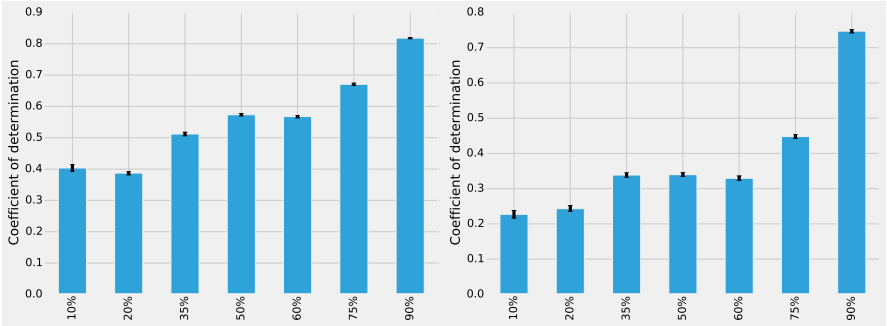


(d) 50% (e) 60% (f) 75%

Fig. 46.: Distribution of scaled feature importance values - 500 network size at different perturbation levels. Each feature is measured across 100 runs.



(a) 100 (b) 200 (c) 300



(d) 400 (e) 500

Fig. 47.: Distribution of coefficient of determination before feature reduction - all network sizes at different perturbation levels. Each feature is measured across 100 runs.

Table 8.: Mean and standard deviation (STD) of features - 200 network size at under different perturbation levels.

Feature	Mean at 10%	STD at 10%	Mean at 20%	STD at 20%	Mean at 35%	STD at 35%	Mean at 50%	STD at 50%	Mean at 60%	STD at 60%	Mean at 75%	STD at 75%	Mean at 90%	STD at 90%
Density	0.0101282659	5.33E-06	0.0101281895	5.32E-06	0.0101261143	5.29E-06	0.0100895329	5.05E-06	0.0100302096	4.42E-06	0.0097255483	2.43E-06	0.0098015164	4.92E-07
Average shortest path (ASP)	0.0370568377	0.0005416057	0.0370574966	0.0005417399	0.0370329189	0.0005430349	0.03706671	0.0005539028	0.0364269036	0.0005888186	0.0336975917	0.0006561388	0.0056761196	0.000148312
Average degree centrality (ADC)	0.0202565319	2.13E-05	0.0202563379	2.13E-05	0.0202522285	2.12E-05	0.0201790658	2.02E-05	0.0200604193	1.77E-05	0.0194510967	9.73E-06	0.0196030329	1.97E-06
Transcription Factors Percentage (TFP)	0.1119141039	0.0005243055	0.1119141039	0.0005243055	0.1119141039	0.0005243055	0.1119141039	0.0005243055	0.1119141039	0.0005243055	0.1119141039	0.0005243055	0.1119141039	0.0005243055
Genes Percentage (GP)	0.8880858961	0.0005243055	0.8880858961	0.0005243055	0.8880858961	0.0005243055	0.8880858961	0.0005243055	0.8880858961	0.0005243055	0.8880858961	0.0005243055	0.8880858961	0.0005243055
Source to Sink Edges Percentage (SSEP)	0.9424244433	0.0006016053	0.9424244433	0.0006016053	0.9424244433	0.0006016053	0.9424244433	0.0006016053	0.9424244433	0.0006016053	0.9424244433	0.0006016053	0.9424244433	0.0006016053
FFL Direct Edges Percentage (FFLDEP)	0.3327104153	0.0242967601	0.3327104153	0.0242967601	0.3327104153	0.0242967601	0.3327104153	0.0242967601	0.3327104153	0.0242967601	0.3327104153	0.0242967601	0.3327104153	0.0242967601
FFL InDirect Edges Percentage (FFLIDEP)	0.4057299845	0.0484417396	0.4057299845	0.0484417396	0.4057299845	0.0484417396	0.4057299845	0.0484417396	0.4057299845	0.0484417396	0.4057299845	0.0484417396	0.4057299845	0.0484417396
Number of Direct FFL edges in Successful Paths (FFLDSPATH)	0.9420705768	0.0021728383	0.9419774408	0.0021692073	0.9396491059	0.0021682078	0.9235567383	0.001992812	0.888852269	0.0017608373	0.764511802	0.0025467154	0.3954096319	0.0027258337
Number of InDirect FFL edges in Successful Paths (FFLIDSPATH)	0.3648145198	0.0044534126	0.5115975862	0.0043780992	0.58620678	0.0066611157	0.5343017008	0.0092510225	0.4403368874	0.0098295612	0.2395796511	0.0060387358	0.0391085798	0.0009909692
Number of Direct FFL edge Occurrences in Successful Paths (FFLDOSPATH)	0.3136061427	0.0287096429	0.3236822132	0.0297262982	0.3412253088	0.0311687097	0.3575154885	0.0331256222	0.3695873754	0.03389353	0.381995133	0.0347679174	0.3765721046	0.031772063
Number of InDirect FFL edge Occurrences in Successful Paths (FFLIDOSPATH)	0.0935705253	0.0171658403	0.1417176583	0.0303269633	0.2020858625	0.0446933631	0.2506756115	0.049743795	0.2903286947	0.054671023	0.3240365494	0.0505046605	0.3932389093	0.0900057405
FFL edge Density (FFLD)	0.4867362044	0.0304602754	0.4867362044	0.0304602754	0.4867362044	0.0304602754	0.4867362044	0.0304602754	0.4867362044	0.0304602754	0.4867362044	0.0304602754	0.4867362044	0.0304602754
FFL Direct Edge Density (FFLDED)	0.412690169	0.0404134623	0.412690169	0.0404134623	0.412690169	0.0404134623	0.412690169	0.0404134623	0.412690169	0.0404134623	0.412690169	0.0404134623	0.412690169	0.0404134623
FFL Source to Sink edge Percentage Density (FFLSSPD)	0.4407567087	0.046371309	0.4407567087	0.046371309	0.4407567087	0.046371309	0.4407567087	0.046371309	0.4407567087	0.046371309	0.4407567087	0.046371309	0.4407567087	0.046371309

Table 9.: Mean and standard deviation (STD) of features - 300 network size at under different perturbation levels.

Feature	Mean at 10%	STD at 10%	Mean at 20%	STD at 20%	Mean at 35%	STD at 35%	Mean at 50%	STD at 50%	Mean at 60%	STD at 60%	Mean at 75%	STD at 75%	Mean at 90%	STD at 90%
Density	0.0062212688	1.36E-06	0.0062211142	1.36E-06	0.0062225985	1.36E-06	0.00622347	1.33E-06	0.0062169454	1.21E-06	0.006198555	7.49E-07	0.00667142	2.12E-07
Average shortest path (ASP)	0.023051944	0.0002017471	0.0239487262	0.0002016273	0.0239391847	0.0002036767	0.0238458697	0.0002100474	0.0233476578	0.0002395686	0.0188153492	0.0002804784	0.0022707031	4.35E-05
Average degree centrality (ADC)	0.0124425375	5.45E-06	0.0124422284	5.45E-06	0.0124451971	5.43E-06	0.01244694	5.30E-06	0.0124338909	4.85E-06	0.0123971101	2.99E-06	0.0133428401	8.47E-07
Transcription Factors Percentage (TFP)	0.1001880878	0.0003725642	0.1001880878	0.0003725642	0.1001880878	0.0003725642	0.1001880878	0.0003725642	0.1001880878	0.0003725642	0.1001880878	0.0003725642	0.1001880878	0.0003725642
Genes Percentage (GP)	0.8998119122	0.0003725642	0.8998119122	0.0003725642	0.8998119122	0.0003725642	0.8998119122	0.0003725642	0.8998119122	0.0003725642	0.8998119122	0.0003725642	0.8998119122	0.0003725642
Source to Sink Edges Percentage (SSEP)	0.946264033	0.0003224314	0.946264033	0.0003224314	0.946264033	0.0003224314	0.946264033	0.0003224314	0.946264033	0.0003224314	0.946264033	0.0003224314	0.946264033	0.0003224314
FFL Direct Edges Percentage (FFLDEP)	0.3076608043	0.0109536667	0.3076608043	0.0109536667	0.3076608043	0.0109536667	0.3076608043	0.0109536667	0.3076608043	0.0109536667	0.3076608043	0.0109536667	0.3076608043	0.0109536667
FFL InDirect Edges Percentage (FFLIDEP)	0.3859057131	0.0359007236	0.3859057131	0.0359007236	0.3859057131	0.0359007236	0.3859057131	0.0359007236	0.3859057131	0.0359007236	0.3859057131	0.0359007236	0.3859057131	0.0359007236
Number of Direct FFL edges in Successful Paths (FFLDSPATH)	0.9373145326	0.0017533299	0.9371947761	0.001754228	0.9346354641	0.0016976064	0.9207208255	0.0015036046	0.8904020862	0.0012799241	0.7695325749	0.0013353439	0.4095672095	0.0012321587
Number of InDirect FFL edges in Successful Paths (FFLIDSPATH)	0.3592552779	0.0020009022	0.5109681145	0.0034837894	0.5900095474	0.0045101193	0.5461942744	0.006134092	0.4544313222	0.0070397346	0.240263831	0.0042262444	0.0461339444	0.000578356
Number of Direct FFL edge Occurrences in Successful Paths (FFLDOSPATH)	0.2886513494	0.0142068617	0.2978697918	0.0143252099	0.3165164606	0.0147182657	0.3328331127	0.0148971104	0.3440122155	0.015107734	0.3552250186	0.014610904	0.3571659487	0.0145741154
Number of InDirect FFL edge Occurrences in Successful Paths (FFLIDOSPATH)	0.0736168824	0.0085290229	0.1204414967	0.0177534107	0.1808980471	0.0313938983	0.2337027057	0.0373952948	0.2692536167	0.0405470201	0.3130497998	0.0393749822	0.3907163503	0.0598646033
FFL edge Density (FFLD)	0.4570814087	0.0114881418	0.4570814087	0.0114881418	0.4570814087	0.0114881418	0.4570814087	0.0114881418	0.4570814087	0.0114881418	0.4570814087	0.0114881418	0.4570814087	0.0114881418
FFL Direct Edge Density (FFLDED)	0.3756928951	0.01733617	0.3756928951	0.01733617	0.3756928951	0.01733617	0.3756928951	0.01733617	0.3756928951	0.01733617	0.3756928951	0.01733617	0.3756928951	0.01733617
FFL Source to Sink edge Percentage Density (FFLSSPD)	0.3972142931	0.0189682198	0.3972142931	0.0189682198	0.3972142931	0.0189682198	0.3972142931	0.0189682198	0.3972142931	0.0189682198	0.3972142931	0.0189682198	0.3972142931	0.0189682198

Table 10.: Mean and standard deviation (STD) of features - 400 network size at under different perturbation levels.

Feature	Mean at 10%	STD at 10%	Mean at 20%	STD at 20%	Mean at 35%	STD at 35%	Mean at 50%	STD at 50%	Mean at 60%	STD at 60%	Mean at 75%	STD at 75%	Mean at 90%	STD at 90%
Density	0.0046463426	6.75E-07	0.0046460843	6.75E-07	0.0046467724	6.70E-07	0.0046478664	6.51E-07	0.0046484416	5.96E-07	0.0046266443	3.79E-07	0.0050152196	1.01E-07
Average shortest path (ASP)	0.0180682189	0.0001100938	0.0189673126	0.0001009941	0.0189065708	0.000102271	0.0184321169	0.0001130311	0.0173984363	0.0001406733	0.0123818807	0.0001840856	0.0009062638	1.63E-05
Average degree centrality (ADC)	0.0092926851	2.70E-06	0.0092921685	2.70E-06	0.0092935448	2.68E-06	0.0092957327	2.60E-06	0.0092968832	2.38E-06	0.0092532885	1.51E-06	0.0100304393	4.02E-07
Transcription Factors Percentage (TFP)	0.0956652361	0.0003056599	0.0956652361	0.0003056599	0.0956652361	0.0003056599	0.0956652361	0.0003056599	0.0956652361	0.0003056599	0.0956652361	0.0003056599	0.0956652361	0.0003056599
Genes Percentage (GP)	0.9043347639	0.0003056599	0.9043347639	0.0003056599	0.9043347639	0.0003056599	0.9043347639	0.0003056599	0.9043347639	0.0003056599	0.9043347639	0.0003056599	0.9043347639	0.0003056599
Source to Sink Edges Percentage (SSEP)	0.9500443786	0.0002460127	0.9500443786	0.0002460127	0.9500443786	0.0002460127	0.9500443786	0.0002460127	0.9500443786	0.0002460127	0.9500443786	0.0002460127	0.9500443786	0.0002460127
FFL Direct Edges Percentage (FFLDEP)	0.2936357995	0.0069311371	0.2936357995	0.0069311371	0.2936357995	0.0069311371	0.2936357995	0.0069311371	0.2936357995	0.0069311371	0.2936357995	0.0069311371	0.2936357995	0.0069311371
FFL InDirect Edges Percentage (FFLIDEP)	0.360841334	0.0318481256	0.360841334	0.0318481256	0.360841334	0.0318481256	0.360841334	0.0318481256	0.360841334	0.0318481256	0.360841334	0.0318481256	0.360841334	0.0318481256
Number of Direct FFL edges in Successful Paths (FFLDSPATH)	0.9349529662	0.0017755637	0.9346294394	0.0017853848	0.9327389907	0.001779858	0.9198081508	0.0017022818	0.8938178538	0.0015970547	0.77916088	0.0013203421	0.409973996	0.0012025696
Number of InDirect FFL edges in Successful Paths (FFLIDSPATH)	0.3739241743	0.0014385691	0.5285926039	0.0022090004	0.6040580587	0.0038111501	0.5644651333	0.0058005423	0.4770484476	0.0070746351	0.2660803591	0.0050207141	0.045572695	0.0005414306
Number of Direct FFL edge Occurrences in Successful Paths (FFLDOSPATH)	0.2809699175	0.0093644645	0.2925608382	0.0086646781	0.3092628032	0.0077778244	0.3286314098	0.0073982097	0.3396236885	0.007181286	0.3443320952	0.0073660484	0.3409120462	0.0083056506
Number of InDirect FFL edge Occurrences in Successful Paths (FFLIDOSPATH)	0.0680188098	0.0081101254	0.1126474118	0.0164404631	0.1647789816	0.025967293	0.216156157	0.0328798341	0.2454855899	0.0342732863	0.2954298517	0.032955701	0.3567701936	0.0537551305
FFL edge Density (FFLD)	0.436133854	0.0103889388	0.436133854	0.0103889388	0.436133854	0.0103889388	0.436133854	0.0103889388	0.436133854	0.0103889388	0.436133854	0.0103889388	0.436133854	0.0103889388
FFL Direct Edge Density (FFLDED)	0.3560644099	0.0093909769	0.3560644099	0.0093909769	0.3560644099	0.0093909769	0.3560644099	0.0093909769	0.3560644099	0.0093909769	0.3560644099	0.0093909769	0.3560644099	0.0093909769
FFL Source to Sink edge Percentage Density (FFLSSPD)	0.3742295134	0.0097371889	0.3742295134	0.0097371889	0.3742295134	0.0097371889	0.3742295134	0.0097371889	0.3742295134	0.0097371889	0.3742295134	0.0097371889	0.3742295134	0.0097371889

Table 11.: Mean and standard deviation (STD) of features - 500 network size at under different perturbation levels.

Feature	Mean at 10%	STD at 10%	Mean at 20%	STD at 20%	Mean at 35%	STD at 35%	Mean at 50%	STD at 50%	Mean at 60%	STD at 60%	Mean at 75%	STD at 75%	Mean at 90%	STD at 90%
Density	0.003743207	3.84E-07	0.0037469121	3.97E-07	0.003744793	3.96E-07	0.0037338029	3.84E-07	0.0037251903	3.60E-07	0.0037113202	2.30E-07	0.0041664741	1.06E-07
Average shortest path (ASP)	0.0096331775	7.99E-05	0.0097146137	8.01E-05	0.0096803428	8.05E-05	0.0089164783	8.22E-05	0.0073056009	7.19E-05	0.0035478256	5.37E-05	0.0001272936	1.42E-06
Average degree centrality (ADC)	0.0074686415	1.54E-06	0.0074938241	1.59E-06	0.0074589586	1.58E-06	0.0074676058	1.53E-06	0.0074503806	1.44E-06	0.0074226405	9.20E-07	0.0083329481	4.24E-07
Transcription Factors Percentage (TFP)	0.0973524684	0.000209932	0.0973177471	0.0002091796	0.0973177471	0.0002091796	0.0973177471	0.0002091796	0.0973177471	0.0002091796	0.0973177471	0.0002091796	0.0973177471	0.0002091796
Genes Percentage (GP)	0.9026475316	0.000209932	0.9026822529	0.0002091796	0.9026822529	0.0002091796	0.9026822529	0.0002091796	0.9026822529	0.0002091796	0.9026822529	0.0002091796	0.9026822529	0.0002091796
Source to Sink Edges Percentage (SSEP)	0.9503615947	0.0001648742	0.9503998831	0.0001639272	0.9503998831	0.0001639272	0.9503998831	0.0001639272	0.9503998831	0.0001639272	0.9503998831	0.0001639272	0.9503998831	0.0001639272
FFL Direct Edges Percentage (FFLDEP)	0.2877973011	0.0086393445	0.2885331128	0.0088325548	0.2885331128	0.0088325548	0.2885331128	0.0088325548	0.2885331128	0.0088325548	0.2885331128	0.0088325548	0.2885331128	0.0088325548
FFL InDirect Edges Percentage (FFLIDEP)	0.3412775178	0.0198585863	0.3419949506	0.0198167909	0.3419949506	0.0198167909	0.3419949506	0.0198167909	0.3419949506	0.0198167909	0.3419949506	0.0198167909	0.3419949506	0.0198167909
Number of Direct FFL edges in Successful Paths (FFLDSPATH)	0.8791536923	0.0079157161	0.8812921059	0.0072428312	0.8766208988	0.0075665767	0.8531760284	0.0092332891	0.8237746906	0.0098632073	0.6975529022	0.0112493938	0.3612801397	0.0046528756
Number of InDirect FFL edges in Successful Paths (FFLIDSPATH)	0.3387563681	0.0024116637	0.4767824709	0.0040657509	0.5380006734	0.0060312976	0.49990674	0.0065882516	0.4157241562	0.0057982786	0.2216269354	0.0033465038	0.0388199192	0.0003861267
Number of Direct FFL edge Occurrences in Successful Paths (FFLDOSPATH)	0.2667275881	0.0064073993	0.2794575629	0.00618499	0.2957932208	0.0051712638	0.3156789618	0.0044493569	0.3270474498	0.0043811476	0.3337160542	0.0043408458	0.3254285622	0.0062987348
Number of InDirect FFL edge Occurrences in Successful Paths (FFLIDOSPATH)	0.0434447229	0.0019989235	0.0774370522	0.0050521145	0.1238624521	0.0103187942	0.1708778386	0.0140421093	0.2035004227	0.0172920544	0.2516273368	0.0212381296	0.3054840744	0.0316843643
FFL edge Density (FFLD)	0.4247097409	0.0119757995	0.4253153947	0.0121345015	0.4253153947	0.0121345015	0.4253153947	0.0121345015	0.4253153947	0.0121345015	0.4253153947	0.0121345015	0.4253153947	0.0121345015
FFL Direct Edge Density (FFLDED)	0.3499201757	0.012229098	0.3507249231	0.0125090863	0.3507249231	0.0125090863	0.3507249231	0.0125090863	0.3507249231	0.0125090863	0.3507249231	0.0125090863	0.3507249231	0.0125090863
FFL Source to Sink edge Percentage Density (FFLSSPD)	0.367414175	0.0126037947	0.3682240686	0.01288717	0.3682240686	0.01288717	0.3682240686	0.01288717	0.3682240686	0.01288717	0.3682240686	0.01288717	0.3682240686	0.01288717

REFERENCES

[1] R. Albert, H. Jeong, and A.L. Barabasi. "Error and attack tolerance of complex networks". In: *Nature* 406 (2000), p. 378.

[2] Albert-László Barabási and Réka Albert. "Emergence of scaling in random networks". In: *science* 286.5439 (1999), pp. 509–512.

[3] Archana Belle et al. "Quantification of protein half-lives in the budding yeast proteome". In: *Proceedings of the National Academy of Sciences* 103.35 (2006), pp. 13004–13009. DOI: `10.1073/pnas.0605420103`. eprint: `http://www.pnas.org/content/103/35/13004.full.pdf+html`. URL: `http://www.pnas.org/content/103/35/13004.abstract`.

[4] John Adrian Bondy and Uppaluri Siva Ramachandra Murty. *Graph theory with applications*. Vol. 6. Macmillan London, 1976.

[5] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[6] Hau Chan, Leman Akoglu, and Hanghang Tong. "Make it or break it: Manipulating robustness in large networks". In: *Proceedings of the 2014 SIAM Data Mining Conference*. SIAM. 2014, pp. 325–333.

[7] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM: a library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 2.3 (2011), p. 27.

[8] Fan Chung et al. "Duplication models for biological networks". In: *Journal of computational biology* 10.5 (2003), pp. 677–687.

[9] Francis S Collins, Michael Morgan, and Aristides Patrinos. "The Human Genome Project: lessons from large-scale biology". In: *Science* 300.5617 (2003), pp. 286–290.

[10] Manlio De Domenico et al. "Navigability of interconnected networks under random failures". In: *Proceedings of the National Academy of Sciences* 111.23 (2014), pp. 8351–8356.

[11] Ernesto Estrada and Naomichi Hatano. "Communicability in complex networks". In: *Physical Review E* 77.3 (2008), p. 036111.

[12] Michael R Fellows et al. "Sharp tractability borderlines for finding connected motifs in vertex-colored graphs". In: (2007), pp. 340–351.

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

[14] Tamar Friedlander et al. "Mutation rules and the evolution of Sparseness and Modularity in Biological Systems". In: *PloS one* 8.8 (2013), e70444.

[15] Preetam Ghosh et al. "Discrete diffusion models to study the effects of Mg2+ concentration on the PhoPQ signal transduction system". In: *BMC genomics* 11.Suppl 3 (2010), S3.

[16] Preetam Ghosh et al. "Principles of genomic robustness inspire fault-tolerant WSN topologies: a network science based case study". In: *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*. IEEE. 2011, pp. 160–165.

[17] S Ghosh et al. "A Discrete Event Based Stochastic Simulation Platform for In silicoStudy of Molecular-level Cellular Dynamics". In: *J Biotechnol Biomaterial S* 6 (2011), p. 2.

[18] Ertan Gul, Baris Atakan, and Ozgur B Akan. "NanoNS: A nanoscale network simulator framework for molecular communications". In: *Nano Communication Networks* 1.2 (2010), pp. 138–156.

[19] B Han, J Leblet, and G Simon. "Query range problem in wireless sensor networks". In: *Communications Letters, IEEE* 13.1 (2009), pp. 55 –57. DOI: 10.1109/LCOMM.2009.081546.

[20] Marti A. Hearst et al. "Support vector machines". In: *Intelligent Systems and their Applications, IEEE* 13.4 (1998), pp. 18–28.

[21] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. *A practical guide to support vector classification*. 2003.

[22] Information-Sciences Institute. *NS-2*. http://isi.edu.nsnam/ns.

[23] M. Isalan et al. "Evolvability and hierarchy in rewired bacterial gene networks". In: *Nature* 452 (2008), p. 840.

[24] Bhanu K. Kamapantula et al. "Dynamical Impacts from Structural Redundancy of Transcriptional Motifs in Gene-regulatory Networks". In: *Proceedings of the 8th International Conference on Bioinspired Information and Communications Technologies*. BICT '14. Boston, Massachusetts, 2014, pp. 199–206. ISBN: 978-1-63190-053-2. DOI: 10.4108/icst.bict.2014.257928. URL: http://dx.doi.org/10.4108/icst.bict.2014.257928.

[25] Bhanu K. Kamapantula et al. "Feature Ranking in Transcriptional Networks: Packet Receipt As a Dynamical Metric". In: *Proceedings of the 8th International Conference on Bioinspired Information and Communications Technologies*. BICT '14. Boston, Massachusetts, 2014, pp. 1–8. ISBN: 978-1-63190-053-2.

DOI: 10.4108/icst.bict.2014.257930. URL: http://dx.doi.org/10.4108/icst.bict.2014.257930.

[26] Bhanu K Kamapantula et al. "Leveraging the robustness of genetic networks: a case study on bio-inspired wireless sensor network topologies". In: *Journal of Ambient Intelligence and Humanized Computing* (2012), pp. 1–17.

[27] Bhanu K Kamapantula et al. "Leveraging the robustness of genetic networks: a case study on bio-inspired wireless sensor network topologies". In: *Journal of Ambient Intelligence and Humanized Computing* (2014), pp. 1–17.

[28] Bhanu K Kamapantula et al. "Performance of wireless sensor topologies inspired by E. coli genetic networks". In: *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*. IEEE. 2012, pp. 302–307.

[29] Bhanu K Kamapantula et al. "Performance of wireless sensor topologies inspired by E. coli genetic networks". In: *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*. IEEE. 2012, pp. 302–307.

[30] Bhanu K Kamapantula et al. "Quantifying robustness of biological networks using NS-2". In: Modeling, Methodologies and Tools for Molecular and Nanoscale Communications. Springer (accepted, to appear), 2014. Chap. N/A.

[31] Guy Karlebach and Ron Shamir. "Modelling and analysis of gene regulatory networks". In: *Nature Reviews Molecular Cell Biology* 9.10 (2008), pp. 770–780.

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

[33] Hiroaki Kitano. "Computational systems biology". In: *Nature* 420.6912 (2002), pp. 206–210.

[34] Hiroaki Kitano. "Towards a theory of biological robustness". In: *Molecular systems biology* 3.1 (2007).

[35] Paul L Krapivsky, Sidney Redner, and Francois Leyvraz. "Connectivity of growing random networks". In: *Physical review letters* 85.21 (2000), p. 4629.

[36] Vito Latora and Massimo Marchiori. *The architecture of complex systems*. Oxford UP, 2004.

[37] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-László Barabási. "Controllability of complex networks". In: *Nature* 473.7346 (2011), pp. 167–173.

[38] Jeantine E Lunshof et al. "Personal genomes in progress: from the human genome project to the personal genome project". In: *Dialogues in clinical neuroscience* 12.1 (2010), p. 47.

[39] Derya Malak and Ozgur B Akan. "Molecular communication nanonetworks inside human body". In: *Nano Communication Networks* 3.1 (2012), pp. 19–35.

[40] Shmoolik Mangan and Uri Alon. "Structure and function of the feed-forward loop network motif". In: *Proceedings of the National Academy of Sciences* 100.21 (2003), pp. 11980–11985.

[41] Michael Mayo et al. "Motif Participation by Genes in E. coli Transcriptional Networks". In: *Frontiers in Physiology* 3.357 (2012). ISSN: 1664-042X. DOI: `10.3389/fphys.2012.00357`. URL: `http://www.frontiersin.org/fractal_physiology/10.3389/fphys.2012.00357/abstract`.

[42] Steven McCanne et al. *Network simulator ns-2*. 1997.

[43] Ron Milo et al. “Network motifs: simple building blocks of complex networks”. In: *Science* 298.5594 (2002), pp. 824–827.

[44] Tadashi Nakano et al. “Molecular communication and networking: Opportunities and challenges”. In: *NanoBioscience, IEEE Transactions on* 11.2 (2012), pp. 135–148.

[45] Saket Navlakha et al. “Topological properties of robust biological and computational networks”. In: *Journal of The Royal Society Interface* 11.96 (2014), p. 20140283.

[46] Alex KS Ng and Janet Efstathiou. “Structural robustness of complex networks”. In: *Physical Review* 3 (2006), pp. 175–188.

[47] NIH. *Cells and DNA - Genetics Home Reference*. `http://ghr.nlm.nih.gov/handbook/basics?show=all`. 2013.

[48] Romualdo Pastor-Satorras et al. “Epidemic processes in complex networks”. In: *arXiv preprint arXiv:1408.2701* (2014).

[49] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[50] Nicolás Peláez and Richard W Carthew. “Biological robustness and the role of microRNAs: a network perspective”. In: *Current topics in developmental biology* 99 (2012), p. 237.

[51] José Antonio de la Peña, Ivan Gutman, and Juan Rada. “Estimating the Estrada index”. In: *Linear Algebra and its Applications* 427.1 (2007), pp. 70–76.

[52] G Piro et al. "Simulating Wireless Nano Sensor Networks in the NS-3 platform". In: *in Proc. of Workshop on Performance Analysis and Enhancement of Wireless Networks, PAEWN, Barcelona, Spain*. 2013.

[53] Robert J Prill, Pablo A Iglesias, and Andre Levchenko. "Dynamic properties of network motifs contribute to biological network organization". In: *PLoS biology* 3.11 (2005), e343.

[54] Software Foundation Python. *Core Python Programming*. `www.python.org`. 1991.

[55] Heladia Salgado et al. "RegulonDB (version 4.0): transcriptional regulation, operon organization and growth conditions in Escherichia coli K-12". In: *Nucleic Acids Research* 32.suppl 1 (2004), pp. D303–D306.

[56] Michael S Samoilov and Adam P Arkin. "Deviant effects in molecular reaction pathways". In: *Nature biotechnology* 24.10 (2006), pp. 1235–1240.

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

[58] Daniel A Schult and PJ Swart. "Exploring network structure, dynamics, and function using NetworkX". In: *Proceedings of the 7th Python in Science Conferences (SciPy 2008)*. Vol. 2008. 2008, pp. 11–16.

[59] Paul Shannon et al. "Cytoscape: a software environment for integrated models of biomolecular interaction networks". In: *Genome research* 13.11 (2003), pp. 2498–2504.

[60] S.S. Shen-Orr et al. "Network motifs in the transcriptional regulation network of Escherichia coli". In: *Nat. Genet.* 31.1 (2002), pp. 64–68.

[61] Khajamoinuddin Syed et al. "Abundance of connected motifs in transcriptional networks, a case study using random forests regression". In: BICT '15 (2015).

[62] Paolo Tieri et al. "Network, degeneracy and bow tie integrating paradigms and architectures to grasp the complexity of the immune system". In: *Theor Biol Med Model* 7.32.10 (2010), p. 1186.

[63] Guido Van Rossum et al. "Python Programming Language." In: *USENIX Annual Technical Conference*. 2007.

[64] Alexei Vázquez et al. "Modeling of protein interaction networks". In: *Complexus* 1.1 (2002), pp. 38–44.

[65] Duncan J Watts and Steven H Strogatz. "Collective dynamics of small-worldnetworks". In: *nature* 393.6684 (1998), pp. 440–442.

[66] Bernard P Zeigler, Herbert Praehofer, Tag Gon Kim, et al. *Theory of modeling and simulation*. Vol. 19. John Wiley New York, 1976.

[67] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. "GloMoSim: a library for parallel simulation of large-scale wireless networks". In: *Parallel and Distributed Simulation, 1998. PADS 98. Proceedings. Twelfth Workshop on*. IEEE. 1998, pp. 154–161.

[68] Zostera. *django-bootstrap3*. `https://github.com/dyve/django-bootstrap3`.