Fall 2019

Structure and topology of transcriptional regulatory networks and their applications in bio-inspired networking

Satyaki Roy

Department: Computer Science

STRUCTURE AND TOPOLOGY OF TRANSCRIPTIONAL REGULATORY NETWORKS AND THEIR APPLICATIONS IN BIO-INSPIRED NETWORKING

by

SATYAKI ROY

A DISSERTATION

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

2019

Approved by

Sajal K. Das, Advisor
Simone Silvestri
Yanjie Fu
Zhaozheng Yin
Dipak Barua

ABSTRACT

Biological networks carry out vital functions necessary for sustenance despite environmental adversities. Transcriptional Regulatory Network (TRN) is one such biological network that is formed due to the interaction between proteins, called Transcription Factors (TFs), and segments of DNA, called genes. TRNs are known to exhibit functional robustness in the face of perturbation or mutation: a property that is proven to be a result of its underlying network topology. In this thesis, we first propose a three-tier topological characterization of TRN to analyze the interplay between the significant graph-theoretic properties of TRNs such as scale-free out-degree distribution, low graph density, small world property and the abundance of subgraphs called *motifs*. Specifically, we pinpoint the role of a certain three-node motif, called Feed Forward Loop (FFL) motif in topological robustness as well as information spread in TRNs.

With the understanding of the TRN topology, we explore its potential use in design of fault-tolerant communication topologies. To this end, we first propose an edge rewiring mechanism that remedies the vulnerability of TRNs to the failure of well-connected nodes, called *hubs*, while preserving its other significant graph-theoretic properties. We apply the rewired TRN topologies in the design of wireless sensor networks that are less vulnerable to targeted node failure. Similarly, we apply the TRN topology to address the issues of robustness and energy-efficiency in the following networking paradigms: robust yet energy-efficient delay tolerant network for post disaster scenarios, energy-efficient data-collection framework for smart city applications and a data transfer framework deployed over a fog computing platform for collaborative sensing.

ACKNOWLEDGMENTS

I express my gratitude to my advisor, Dr. Sajal K. Das, for his advice, support and guidance throughout my doctoral studies at Missouri University of Science and Technology. I am thankful to my parents, Dr. Syamal Roy and Dr. Amrita Dutta, as they have been true pillars of strength. I must specially mention of three of closest friends, Nitish Uplavikar, Sangeeta Sur and Sneha Mitra, who helped me immensely whenever I reached out to them.

I am very grateful to my colleagues and co-authors Dr. Preetam Ghosh, Dr. Nirnay Ghosh, Dr. Vijay K. Shah, Dr. Md. Aminul Islam, Dr. Simone Silvestri, Dr. Mayank Raj and Dr. Dipak Barua for their inputs. The past and present members of CReWMaN research lab (especially Vijay K. Shah, Rakesh Kumar, Alec Bayliff, Pratool Bharti) as well as Dr. Venkata Sriram Siddhardh Nadendla deserve special mention as they stood by me in testing times.

Finally, I am grateful to my PhD committee members Dr. Yanjie Fu, Dr. Simone Silvestri, Dr. Zhaozheng Yin and Dr. Dipak Barua for their insights that helped me significantly improve the contents and presentation of my dissertation.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Table Page

1. INTRODUCTION

Biological systems are characterized by certain key properties which are a direct consequence of evolution. They adapt to changing environment, exhibit high resilience to failures and attacks, and collaboratively accomplish complex tasks, while making minimum use of available resources. Researchers in different domains of computer science are exploring such properties, which aid the living organisms in combating survival challenges.

In recent years, there have been examples of modeling biological systems in handling communication networks and optimization problems [1]. The first example that comes to mind is that of slime mold *Physarum polycephalum*. The slime mold is a single-celled amoeboid organism which has a tendency of foraging for food sources using shortest paths. This unique characteristic of *Physarum* has been extrapolated in the design of efficient transport networks with minimum average distance between node pairs [2]. Also, researchers tapped into swarm intelligence-based algorithms like the foraging habits of ants (that make use of a chemical called pheromone). The objective behind these algorithms is to conceive distributed systems, where individual components can interact with one another and their environment. Finally, it is known that the immune system of living organisms can detect the slightest aberrations in the environment and learn from past experiences. Taking cue from immune system of mammals, researchers have formulated the Artificial Immune Systems (AIS) which can be utilized to solve computational problems [3].

There have been a few attempts to survey the existing literature in bio-inspired networking, defined as a class of strategies for efficient and scalable networking under uncertain conditions [1]. Dressler et al. discussed the different challenges in the design of next generation network architectures such as scalability, heterogeneity, need for infrastructure-less wireless communications. They delineated the different ways of modeling a computing application on a biological phenomenon, followed by different bio-inspired techniques such as swarm intelligence, ant colony optimization, routing, epidemic spread, activator inhibitor system, etc. Forbes discussed how bio-inspired computing transcends artificial neural networks and genetic algorithms and enters into DNA computing and biological hardware [4]. Kotteeswaran discussed how living cells structure and functioning can be applied in membrane computing, finite automata, etc. [5]. Meisel et al. explored the possible overlaps in biology and computer networks research such as parallelism, innate fault-tolerance [6].

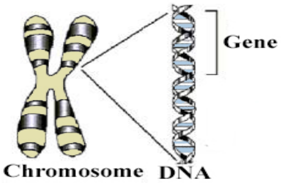


Figure 1.1. Chromosome, DNA and gene.

The robustness of a biological network called transcriptional regulatory network (TRN) in the face of mutation or noise has been a key area of interest in computational biology [7]. Studies show that the robustness of TRNs can be ascribed to its underlying network topology [8]. In this thesis, we make an effort to apply the innate robustness of TRNs in the design of fault-tolerant and energy efficient computer network architectures and protocols by exploiting the various standard graph-theoretic attributes of TRNs such as scale free out-degree distribution, abundance of subgraphs called *motifs*, low graph density, small world property, etc. We discuss such graph-theoretic properties of TRNs, including the implications of its motif abundance. Specifically, we discuss the association between topological robustness of TRNs and its motif abundance, identify the metrics in literature that quantify motifs and discuss how such metrics can lead to smart networking solutions. Let us now provide a brief background of TRNs.

1.1. CHROMOSOME, DNA AND GENE

Genetic material of any living organism is contained within the cell nucleus in thread-like structures called *chromosomes*. These chromosomes are composed of molecules of deoxyribonucleic acid (DNA), and segments of DNA are called *genes* (Figure 1.1). In living cells, protein synthesis takes place thorough a process called *gene expression*. In other words, a gene in DNA is expressed (or "turned on") when it makes the protein it specifies. The first step in gene expression is called *transcription*. It involves copying a gene's DNA sequence to make an RNA molecule. Let us briefly go over the steps in transcription:

- An enzyme called *RNA polymerase* binds to a sequence of DNA called the promoter, found near the beginning of a gene. RNA polymerase then separates the DNA strands, providing the single-stranded template needed for transcription.

- One strand of DNA, the template strand, acts as a template for RNA polymerase. As it "reads" this template one base at a time, the polymerase builds an RNA molecule out of complementary nucleotides.

- Sequences called terminators signal that the RNA transcript is complete. Once they are transcribed, they cause the transcript to be released from the RNA polymerase.

In eukaryotic cells, once the RNA is processed to make final product, called a messenger RNA (or mRNA), a process called *translation* takes place, in which the mRNA is read to build a proteins containing a specific series of amino acids.

1.2. TRANSCRIPTION FACTORS

Transcription factors (TFs) are proteins that regulate the transcription of genes. Transcription factors can be *activators* and *repressors*.

- Activators are TFs that activate transcription. They may help the RNA polymerase to bind to the promoter.

- Repressors impede the process of transcription, Repressors may get in the way of RNA polymerase, preventing them from binding to the binding sites, called promoters.

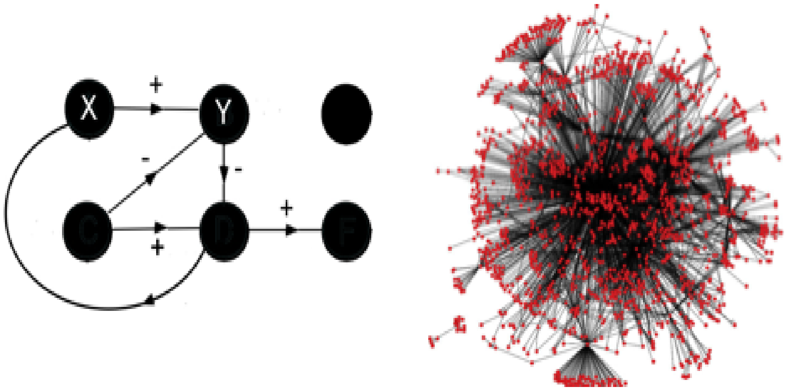


Figure 1.2. Transcriptional Regulatory Network (TRN). Graph theoretic representation of TRN and snapshot of *E. coli* transriptional regulatory network (taken from [9]).

1.3. TRANSCRIPTIONAL REGULATORY NETWORKS

Transcriptional regulatory networks (TRNs) are represented as directed, signed graphs in which nodes represent genes or transcription factors (TFs) and edges correspond to enhancing or inhibitory regulations between TFs and target genes [10]. Positive and negative signs on directed edges of a TRN correspond to enhancing and inhibitory regulation, respectively [11]. Figure 1.2 shows a simple graph representation of the transcriptional regulatory network, where the nodes labeled X is a regulating gene/TF and Y is a regulated gene as well as a snapshot of TRN of a unicellular organism *E. coli*. Moreover, there is evidence to show that the edges between the regulating and target nodes are weighted, and these weights indicate the strength of regulation [12].

1.4. DATASET

The validated and nearly complete TRNs of *E. coli* and *S. cerevisiae* were extracted from *GeneNetWeaver* [11]. The human and mouse TRNs were obtained from the TRRUST database [13, 14]; these two TRNs catalogue the partially known validated interactions between TFs and genes in these two organisms. The orders and sizes of the four TRN topologies considered in this work are summarized below:

Table 1.1. TRN graphs.

TRN type	*E. coli*	*S. cerevisiae*	human	Mouse
No. of nodes	1565	4441	2862	2456
No. of edges	3758	12873	8427	6490

Note that the complete information of the sign (i.e. up or down regulation) and magnitude of influence of TFs on their target genes is not available in these datasets. As mentioned before, we consider TRNs as signed and directed graphs.

1.5. MOTIVATION

Biological systems are characterized by certain key properties which are a direct consequence of evolution. They adapt to the changing environment, exhibit high resilience to failures and attacks, and collaboratively accomplish complex tasks, while making minimum use of available resources. These natural survival mechanisms are a result of millions of years of *evolution*. As a consequence, it is believed that mimicking these properties can lead to effective solutions in various fields of computing.

In recent years, there have been examples of modeling biological systems in handling communication networks and optimization problems [1]. The first example that comes to mind is that of slime mold *Physarum polycephalum*. The slime mold is a single-celled amoeboid organism which has a tendency of foraging for food sources using shortest paths. This unique characteristic of *Physarum* has been extrapolated in the design of efficient transport networks with minimum average distance between node pairs [2]. Also, researchers have tapped into swarm intelligence-based algorithms like the foraging habits of ants (that make use of a chemical called pheromone). The objective behind these algorithms is to conceive distributed systems, where individual components can interact with one another and their environment. Finally, it is known that the immune system of living organisms can detect the slightest aberrations in the environment and learn from past experiences. Taking cue from immune system of mammals, researchers have formulated the Artificial Immune Systems (AIS) which can be utilized to solve computational and mathematical problems [3]. We now turn our focus to the common challenges faced by the large scale communication network topologies. They include:

- *Dynamic nature:* The future networking architectures need to be dynamic w.r.t node behaviors, traffic and bandwidth demand patterns, channel and network conditions.

- *Resource constraints:* As newer services are incorporated in the network, there will exist demands for higher bandwidth capacity and energy overhead.

- *Infrastructure-less architecture:* Centralized solutions for wireless sensor networks, delay tolerant networks, mobile ad hoc networks are no longer viable because of unprecedented growth in network size.

- *Heterogeneous architecture:* Future large-scale networks will possess a large range of network elements, requiring varied levels of modeling and realization of network hierarchies.

- *Component failures:* Most real-world networks are prone to component failures, either due to energy depletion, hardware fault or physical damage.

It is believed to be possible to come up with smart and scalable networking solutions that meet the aforementioned challenges. To this end, there exists this emerging field called *bio-inspired networking*, defined as a class of bio-inspired strategies for efficient and scalable networking under uncertain conditions. The primary goal in bio-inspired networking is to design energy-efficient and *robust* information dissemination systems. In our context, we define robustness as *the ability of the network to carry out information flow despite component (i.e. node or link) failures*.

1.6. RESEARCH GOALS

This thesis addresses two broad research topics: (1) identification of topological attributes of TRNs and (2) apply attributes to design robust, energy-efficient communication network solutions (Figure 1.3).

1.6.1. Structure and Topology. We propose a three-tier topological characterization of TRN to analyze the interplay among the myriad graph-theoretic properties such as scale free out-degree distribution, low graph density, abundance of subgraphs called motifs, preferential attachment, small world property, and TF-gene regulation. We then identify the fatal flaw in TRN topology: vulnerability to failure of well-connected nodes, called *hubs*. In addition, we come up with a computational model that can influence (i.e. activate or deactivate) a specific set of genes in TRN.

1.6.2. Applications in Bio-inspired Networking. We have applied the graph-theoretic properties of TRNs in the design of different networking solutions, such as:

1. *Wireless Sensor Networks (WSN):* We propose an efficient WSN topology that is resilient to random and targeted node failures.

2. *Disaster Response Networks (DRN):* We design an energy-efficient and robust DRN topology that enables seamless communication through a makeshift network in the absence of primary communication infrastructure.

3. *Internet of Things (IoT) networks*: We introduce a TRN-based distributed IoT-based data collection framework for smart city application that combines Quality of Information (QoI) with energy efficiency.

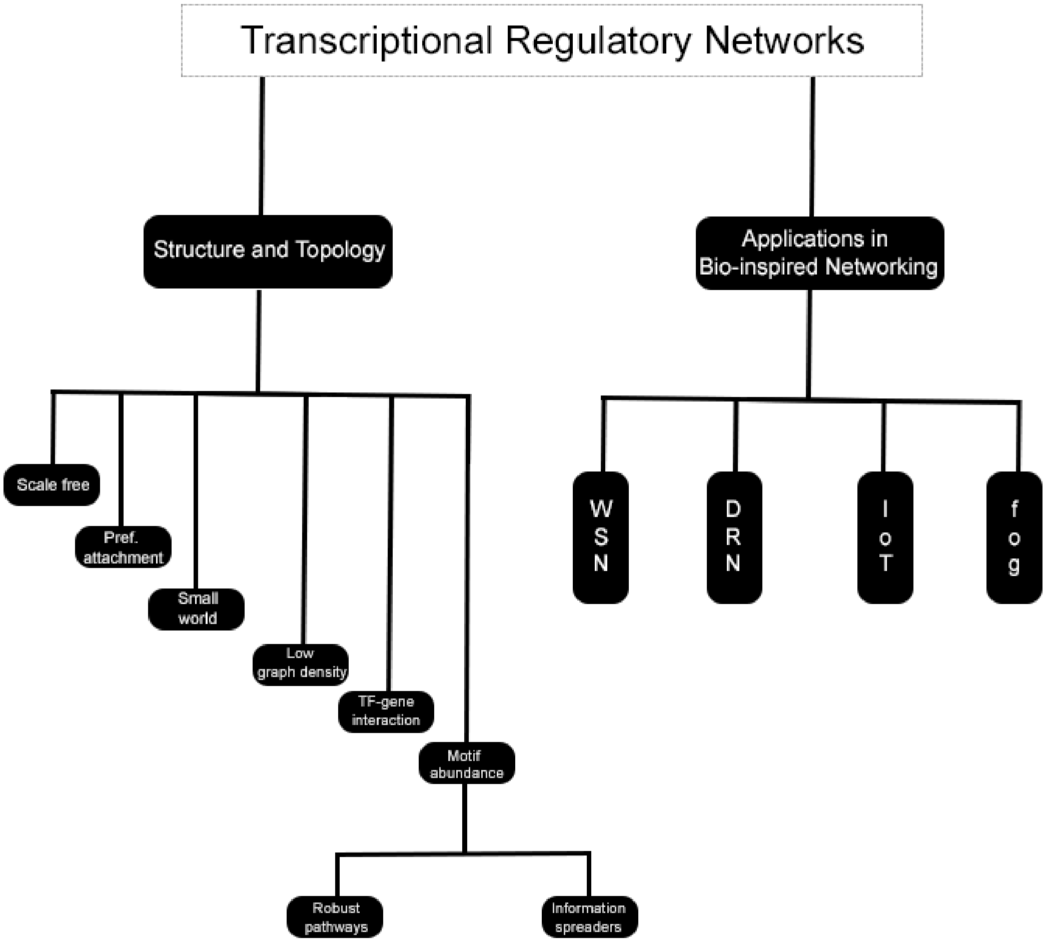


Figure 1.3. Breakdown of the research goals.

4. *Data Transfer Framework in Fog Computing Platforms*: We introduce a data transfer framework that consists of a robust network of fog nodes and mobile devices.

We explore myriad facets of networking while designing topologies for the above networks, such as *source to sink data delivery*, *communication delay*, *network lifetime*, *quality of data reported at the base station*, *regulation of device energy levels*, etc.

1.6.3. Summary of Contribution. Here is a summary of contributions of my dissertation:

- We propose a *three-tier topological characterization* to study the interplay between the varied graph theoretic properties of TRNs such as low graph density, scale free out-degree distribution, abundance of motifs, preferential attachment, TF-gene regulation and robustness against random failure.

- We study how FFL motifs render topological robustness by creating multiple communication pathways as well as acting as the most efficient spreaders of information.

- TRNs can act as templates for design of energy-efficient and robust wireless sensor networks, disaster response networks, IoT-Net as well as fog networks.

- We conceive a computational framework that can effectively identify potential drug targets in TRNs.

- TRNs can further inspire design of several smart networking solutions such as a bio-inspired routing protocol. Furthermore, analysis of the hub-and-spoke architecture and signed FFL motifs in TRNs can lead to new insights into TRN topology.

2. LITERATURE REVIEW

In this section, we briefly discuss the existing literature on the structure and topology of TRN and its application in wireless sensor networks. In addition, we discuss the existing routing protocols in delay tolerant networks and disaster response networks and how it serves as a motivation for the construction of a bio-inspired disaster response network.

2.1. MOTIF ABUNDANCE

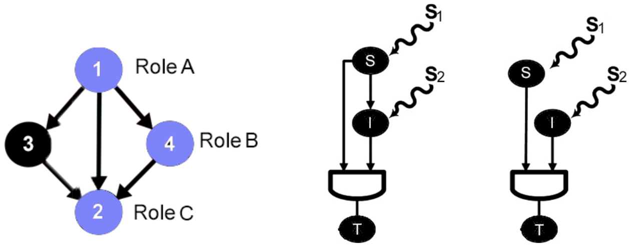
Network motifs are statistically over-represented subgraphs that are simple building blocks of complex networks [15, 16]. Motifs play important functional role in the TRN, like controlling the gene expression by moderating the responses to fluctuating external signals. From a biological standpoint, motifs can be considered to be input-output devices which require inputs like heat, nutrients and pressure and produce outputs like regulation signals that act upon the targets. Based on the duration and intensity of regulation, the motifs could control some of the vital functions in the living organisms [17]. It is noteworthy that the frequency of motifs in complex networks such as TRN when compared to the random network, is particularly high [18, 19, 20, 21] (as we shall discuss shortly after).

Motif detection tools: There exist a variety of tools to detect network motifs. Two popular tools are MFINDER [22] and MAVISTO [23]. While MFINDER is capable of detection of network motifs, MAVISTO is equipped with a visualization tool to capture the presence of a motif in a network by a force-directed graph layout algorithm. Wernicke et al. put forward a scalable and fast motif detection tool, called FANMOD [24], that is capable of handling colored vertices and edges to model different kinds of node interactions such as finding motifs in protein-gene interaction networks. In FANMOD, the subgraphs are grouped into isomorphic subgraph classes based on canonical graph-labeling algorithm

NAUTY [25]. It then calculates the frequency of subgraph classes in a user-specified number of random graphs generated from the original network by switching edges between vertices. More details about other existing motif detection tools can be found in [26].

Let us discuss the most abundant motifs and some of their functional roles.

1. **Feed forward loop (FFL):** The Feed Forward Loop (FFL) is known to be one of the most abundant and significant motifs in the transcriptional regulatory networks [27, 28]. Figure 2.1(a) shows the FFL, where TFs S and I regulate the expression of gene T. S is the *general TF*, I is the *specific TF* and T is the *effector operon*. Evidently S regulates T directly and indirectly (via I). There are two categories of FFLs: (a) coherent (b) incoherent. The sequence of $+/-$ signs will determine whether the FFL is coherent or incoherent. The coherent and incoherent FFLs have different functional roles in the TRN.



(a) (b)

Figure 2.1. Feed Forward Loop. (a) Feed Forward loop (FFL) motif with different FFL motif centrality roles (b) Left: TF S regulates TF I, and S and I jointly regulate T; right: S and I regulate T.

A FFL is called *coherent* if the direct effect of the general TF S on the effector operon T, has the same sign as the indirect effect through the specific TF I. *Incoherent* FFLs have the opposite signs for the two different paths.

It is important to note, while calculating the net effect of activation and inhibition one may make use of basic mathematics rule of product of two signs. If we calculate the product of like-signs (+ and +; − and −) we get a net positive result, analogously, product of unlike signs (+ and −) yield net negative result. This leads to 8 types of FFLs, as depicted in Figure 2.2: four each belonging to coherent and incoherent categories.

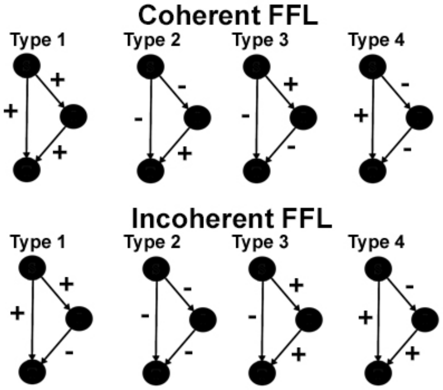


Figure 2.2. Coherent and incoherent FFLs.

The FFL has two input signals, the inducers, S_1 and S_2, which are molecules that activate or inhibit the activity of S and I (Figure 2.1(b) (left)). Depending on whether they are coherent or incoherent, the FFL motifs have specific information-processing roles by regulating the activation of target gene T defined by the time (called *response time*) it takes a gene product to reach its steady-state level [19]. Incoherent FFLs act as accelerators, i.e., they provide a mechanism for speeding up the responses of T, whereas the coherent FFLs lead to delay in the response of T when compared against direct regulation (shown in Figure 2.1(b) (right)).

Abundance of FFL in TRN: As per FANMOD, out of 455152 3-node subgraphs enumerated in human TRN, approximately 5850 motifs are FFLs or motifs possessing FFLs as building blocks. Abundance of FFL exceeds the other triangular motif in TRN, which is the cyclic triangle Feedback Loop (FBL) (as shown in Appendix A Section 1).

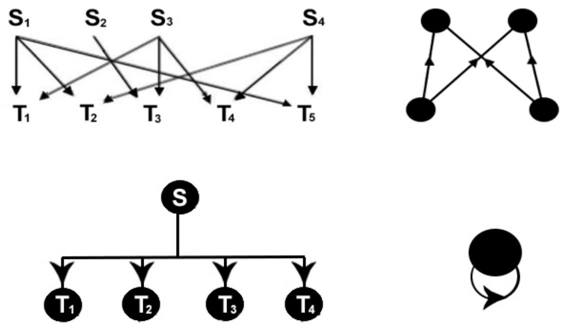


Figure 2.3. Motifs in TRN. Top Left: Dense overlapping regulon, Top right: bi fan, Bottom Left: Simple Input Module, Bottom Right: auto-regulation.

2. **Dense overlapping regulons (DORs)**: These motifs constitute a set of regulating genes S_i and target genes T_i set in the form of a bipartite graph. They are called dense because they are present in cascades or layers as depicted in Figure 2.3(top left). The DORs are responsible for a number of biological functions like carbon utilization, growth and stress response [29]. We often consider a 4-node sub-motif of DOR Figure 2.3(top right) to be a single entity in a cascade of DOR. These are called *bi fan*.

   Abundance of bi fan in TRN: As per FANMOD, out of 43995531 4-node subgraphs enumerated in human TRN, approximately 132481 motifs are bi fans or possess bi fans as building blocks.

3. **Single Input Modules (SIMs):** This is a motif which has a single regulating TF S which regulates a number of genes S_i [30, 31]. The name, single input module explains that there is only one regulator. The key characteristic of this motif is that all the target genes are either activated or all are repressed. Also, the regulating gene is capable of regulating itself. As shown in Figure 2.3 (bottom left), regulator S has a self-loop. The primary biological role of the SIM is to cause the collective expression of multiple genes, even though the regulator may have varying activation threshold for different target genes.

4. **Auto-regulation:** When a gene binds its own promoter (Figure 2.3 (bottom right)) and activates itself we call it positive auto-regulation, and when it represses itself it is called negative auto-regulation [32]. Simulations on boolean network models of TRN have shown that robustness and stability of TRN correlates with frequency of auto-regulation in the network [33]. *E. coli*, *S. cerevisiae*, human and mouse TRNs have 110, 0, 24 and 28 auto-regulation motifs, respectively.

2.2. WIRELESS SENSOR NETWORK

In the domain of wireless communication, there exists specific type of networks, called wireless sensor networks (WSNs), consisting of small, resource-constrained and battery-powered sensor devices that are deployed over an area of interest, to collect and deliver critical information in a wide variety of applications, such as environmental monitoring, health-care, target tracking and disaster management. Sensor nodes collect samples from the environment, process them, and forward the results over multiple hops to the sinks [34]. WSNs exhibit several functional similarities with TRNs [35]. First, the functioning of a WSN is greatly affected by the data sampled from the sensing field, similar to how environmental factors (such as the concentration of chemicals) impact the working of TRNs [36]. Second, communications in WSNs are subject to link failures which depend on multiple factors, like the distance between sensor nodes, temporal fluctuations of the wireless channel and interference. Likewise, TRNs are affected by stochasticity inherent in bio-chemical functioning of cells and organelles, as well as interference generated by alterations in the properties of molecules, for instance, due to molecular crowding [37]. Finally, nodes in WSN can fail due to hardware faults or battery depletion, but the network still remains functional as long as one sink can be reached by remaining nodes. Similarly, TRNs are robust against the random removal of nodes since important nodes remain reachable, despite other node or link failures due to external perturbations (e.g., chemicals) [38]. Such nodes correspond to the attractors in the dynamic state transition space, i.e., the states eventually reached by the system [39], [40] [41]. Just as robustness in TRNs depends on attractors, robustness in WSNs relies on the reachability of sink nodes [42]. Taking the analogies between WSNs and TRNs into consideration, Nazi et al. conceived an approach to design efficient WSN graphs that exploit robust signal transmission properties of TRN. They proposed a bio-inspired node deployment solution in WSNs [43], [44] that mimics the TRN topologies; however such a strategy does not work on randomly deployed WSNs. To address this, attempts were made to construct bio-inspired WSNs, by establishing a rigorous correspondence i.e., a one-to-one mapping between the nodes in a already deployed WSN and TRN graph [45], [46]. Such a mapping was achieved by means of graph embedding under the optimization criterion of minimizing the interference between different nodes. While the resultant bio-inspired WSNs exhibit high packet delivery rate and low network latency, its scalability is greatly constrained by the inherent topological characteristics of the input TRN graph.

There are two other interesting works in the application of TRN in WSN. First, Markham et al. have conceived a target tracking application, called *discrete GRN* (dGRN) that is inspired from the manner in which the cell regulate their behavior based on the local level of protein concentration and protein diffused from neighbor cells [47]. They experimentally show that the proposed framework is particularly beneficial in scenarios where nodes must tune their sampling rates to track a moving target with a certain accuracy. The efficacy of the dGRN framework is evaluated both in a simulation environment, and in a real environment with eight T-Mote Sky nodes tracking a light-emitting target. Second, Byun et al. employed the principles of TRN to design a self-organizing control for WSN that meets both the requirements of energy-efficiency and delay guarantee [48]. Here, each sensor node schedules its state autonomously according to the controlled gene expression and protein concentration of the proposed TRN model. They carried out simulation experiments to show that the proposed approach achieves good performance in meeting delay requirements and conserving energy in WSN systems.

2.3. DISASTER RESPONSE NETWORKS

Existing research in both DRNs and DTNs have primarily focused on intelligent routing protocols for achieving high packet delivery [49] and enhancing energy efficiency [50, 51, 52, 53, 54, 55]. Epidemic routing [49] is the simplest routing that replicates and transmits messages to every encountering node, thereby achieving highest packet delivery, while consuming significant amount of energy. Inter Contact Routing protocol (ICR) [50] attempts to control message replication and transmission by estimating route delays and delivery probabilities, by exploiting recurrent mobility and contact patterns of survivors and responders. Cluster based Topological Routing (CTR) [51] utilizes naturally occurring survivor groups and minimizes the number of data transmissions, by allowing survivor nodes to only communicate with their respective well-connected group representatives, called *exemplars*. PROPHET [52] estimates the high delivery predictability routes to the destinations using history of encounters and transitive property of meeting with nodes. MaxProp [53] calculates the probabilities of message delivery from meeting frequencies and sorts the messages in the transmission buffer accordingly. Spray and Wait [54] initially limits the number of message replications (spray phase); then each node waits for an opportunity of direct message delivery to the destination node (wait phase). Readers may refer to survey articles [56, 57] for a complete understanding on DTN routing protocols.

Recently few topology control approaches [58, 59, 60] have been proposed for DTNs. The authors in [58] and [59] aim to build a sparse structure from the original graph such that (i) the network is connected over time; and (ii) the total energy cost of the structure is minimized. Finally, the authors in [60] aim to construct a spanning tree such that it minimizes the energy cost and satisfies the time delay threshold.

The aforementioned works either attempt to improve packet delivery and/or enhance energy efficiency of the network. However, network robustness against component failures, which is a key requirement for DRNs, has remained largely unaddressed. This motivates us to design a novel bio-DRN topology that addresses both energy efficiency and network robustness, without compromising the desired QoS.

2.4. SMART CITY BASED INTERNET-OF-THINGS (IOT) NETWORK

There are very few works in existing literature that address IoT-based data collection frameworks. Jaiswal *et al.* [61] proposed a model for the sensors to monitor medical data and transfer patient data to gateway. Almeida *et al.*[62] proposed an approach to use the IoT devices to collect and manage data related to the elderly behavioral changes that can potentially be early signs of cognitive impairments. In [63], the authors introduced a IoT-based framework for offloading industrial meter data to the cloud storage for data processing. Capponi *et al.* [64] proposed a distributed data collection mechanism for smart city application.

2.5. MCS-BASED DATA ACQUISITION FRAMEWORKS

These frameworks in MCS can either be application-specific or general-purpose. Application-specific frameworks, such as *GasMobile* [65] and *NoiseMap* [66], are designed to cater to only one type of application at a time. These frameworks have been developed to monitor air and noise pollution, respectively. On the contrary, *general-purpose frameworks*, like Google, have the capability to serve many applications at the same time. *BLISS* [67] implements an online learning algorithm to collect general-purpose data. The framework optimally assigns tasks to the users in lieu of rewards, subject to constraint of a fixed budget. Wang *et. al* [68] proposed an energy efficient algorithm for uploading the sensed data by classifying users into two categories: (i) users who use LTE/4G/3G through data plans with mobile operators and (ii) users who use free-of-charge networks like Wi-Fi or Bluetooth. For the first category, the proposed approach attempts to minimize the energy cost during data uploading. In [69], the authors propose an energy efficient data delivery by piggybacking the sensed data with voice calls. Liu *et al.* [70] define a new routing mechanism for data delivery in MCS systems to counter user selfishness. Data delivery is enabled through opportunistic communications and is forwarded in a delay-tolerant fashion only by cooperative, non-selfish nodes.

Few works have used the paradigm of fog computing in MCS. In *CARDAP* [71] and *CAROMM* [72] the fog platform has been used to perform distributed data analytics. Fiandrino *et al.* [73] exploit the computing capacity of the fog platform for efficient user recruitment and task completion in participatory MCS systems. In contrast, this work uses fog devices for Wi-Fi *GO* selection and energy efficient data transfer to the MCS platform.

2.6. SELECTION OF WI-FI DIRECT GROUP OWNER

This is one of the main functionalities in establishing Wi-Fi direct groups. Some works in literature have proposed various *GO* selection strategies for efficient data sharing among devices in the proximity. *WD2* [74] algorithm automatically selects best *GO* based on the Received Signal Strength Indication (RSSI) measurement. It operates in standard Wi-Fi direct mode by which each device collects the RSSI reading from nearby devices, and a *GO Intent (GI)* value is calculated based on such collected measurements. The devices then exchange their *GI* values during the discovery phase and the one that exposes the highest *GI* value creates the group. In [75], authors account for *GO* selection based on residual energy, implying that candidates for best *GO* will vary with time. In [76], three different approaches to select *GO* were proposed: (i) the device with the highest ID in the surroundings; (ii) the peer that has the shortest average distance from the other nodes; (iii) the node with less mobility with respect to its neighbours. However, considering a single metric is not sufficient to manage the complex dynamic involving mobility of the nodes.

WFD-GM [77] incorporates mobility of nodes and improves the earlier works by defining a suitability index for *GO* selection based on four factors - amount of available resources of the local device (e.g., battery level, free CPU, free memory), the current number of peers discovered in proximity, the capacity of the node (i.e., the number of incoming connections that the device can still accept), and the stability index, which provides a measure of the ability of the node to create a long-lasting group (i.e., a group that will not be rapidly destroyed due to the local node mobility). In [78], authors propose three policies for electing suitable *GO* of Wi-Fi direct groups - *static grouping (SG)*, *point of interest grouping (PG)*, and *dynamic grouping (DG)*. However, in this work, we formulated a fitness score based on mobile device user's activeness property in using the MCS platform, his/her promptness in executing recent tasks, and residual energy of the device to select Wi-Fi direct GO.

3. THREE TIER TOPOLOGICAL CHARACTERIZATION OF TRN

In this section we discuss the key topological properties of a TRN. We utilize the *three tier topological characterization* of a TRN [79, 80] to visualize some of the graph properties. This characterization is a simplified representation of the hierarchical structure of TRNs discussed in existing literature. Gerstein et al. studied the network interactions of different TFs and mRNAs in humans on the basis of properties such as connectivity, motifs, etc. [81]. Similarly, Bhardwaj et al. employed breadth-first search (BFS) to form a hierarchy of TFs based on regulating-regulated TF relationships to identify the master regulators in *E. coli* and *S.cerevisiae* TRNs [82]. Finally, Ma et al. [29] proposed the five-level hierarchy of TFs and operons in *E. coli*.

The three tier topological characterization classifies the TRN nodes into three tiers based on in- and out-degree distribution. The three tiers are:

- **Tier 1** nodes with only out-degree edges
- **Tier 2** nodes with in- and out-degree edges.
- **Tier 3** nodes with only in-degree edges

We discuss the node and edge distribution across the three tiers in *E. coli*, *S. cerevisiae*, human and mouse TRN. This topological characterization illustrates that information flow in the TRN takes place from the **hubs** (high degree TF nodes in tiers 1 and 2) to the *non-hubs* (tier 3 genes).

3.1. NODE AND EDGE DISTRIBUTION

We tabulate the distribution of nodes and edges within and across tiers in Tables 3.1 and 3.2.

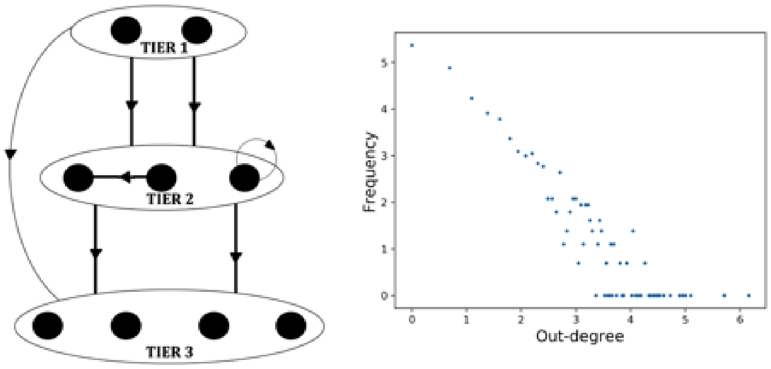


Figure 3.1. Three tier topology and out-degree distribution in TRN. The directed edges indicate potential edges across and within the three tiers (taken from [79, 80]); Right: Out-degree distribution of human TRN on a log log scale.

3.1.1. Node Distribution. Table 3.1 shows that tiers 1 and 2 make up the smaller fraction of nodes in *E. coli*, *S. cerevisiae*, human and mouse TRN. As shown in Figure 3.1 (left), all nodes containing self-loops belong to tier 2. It is noteworthy that tier 3 of TRN holds the vast majority of TRN nodes: 88.6% in *E. coli*, 96.4% in *S. cerevisiae*, 72.2% in human and 66.3% in mouse.

Table 3.1. Percentage of nodes in each tier of *E. coli*, *S. cerevisiae*, human and mouse TRN.

	E. coli	*S. cerevisiae*	human	Mouse
Tier 1	4.1	0.7	12.9	14.8
Tier 2	6.2	2.8	14.8	18.8
Tier 3	89.7	96.4	72.2	66.3

3.1.2. Edge Distribution. The arrows in Figure 3.1(left) are indicative of the possible direction of edges within and across tiers. Only possible edges in TRN are between tiers 1 → 2, 1 → 3, 2 → 2 and 2 → 3. Self-loops, if any, are found in tier 2. We summarize the percentage of edges between each of the tier nodes in Table 3.2. Note that in all TRNs, well over 50% of total edges are between tiers 2 and 3.

Table 3.2. Percentage of edges in each tier pair in *E. coli*, *S. cerevisiae*, human and mouse TRN.

Tier pair	*E. coli*	*S. cerevisiae*	human	Mouse
(1 → 2)	0.5	0.5	4.0	6.2
(1 → 3)	10.3	11.0	11.1	10.5
(2 → 2)	8.0	3.3	18.3	27.8
(2 → 3)	81.0	85.1	66.5	55.3

3.2. GRAPH PROPERTIES

In this section we discuss the following interesting topological properties of TRN: (1) *scale free out-degree distribution*, (2) *low graph density*, (3) *small world property*, (4) *motif abundance*, (5) *clustering tendency*, (6) *robustness to random node failures* and *vulnerability to hub node failures*, (7) TF-gene regulation and (8) *preferential attachment*. Note that the first six of these properties (1 - 6) are visualized using the three tier topological characterization (illustrated in Figure 3.1(left)).

3.2.1. Scale Free Out-degree Distribution. A scale free network is one whose degree distribution follows a powerlaw. (A powerlaw distribution has the functional form $P(k) = Ak^{-\gamma}$. Here, A is a constant that ensures that the $P(k)$ values add up to 1 and the degree exponent γ is usually in the range $2 < \gamma < 3$). Such networks possess a few well-connected nodes, called *hubs*, that have high connectivity, while most of the nodes have a lower degree of connectivity [83, 84].

Table 3.1 shows that tier 1 and tier 2 nodes collectively account for approximately less that 10% of total nodes in *E. coli* and *S. cerevisiae* and less than 35% nodes in human and mouse, but possess all the out-degree edges. Conversely, tier 3 nodes make up most of the nodes in both TRNs but have zero out-degree. Since a few nodes have a disproportionately high out-degree, TRNs are out-degree scale free in nature. In Figure 3.1 (right), we have the degree distribution of human TRN on a log-log scale showing a clear powerlaw.

3.2.2. Low Graph Density. In a directed graph $G(V, E)$, we can define graph density D on a scale of 0 to 1, as:

$$D = \frac{|E|}{|V| \times (|V| - 1)} \tag{3.1}$$

Using the above equation, $D = 1$ indicates a complete directed graph and $D = 0$ corresponds to an empty graph. TRN is also characterized by low graph density [85]. As a validation, we use Eq. 3.1 to show that the graph density of TRN is very low (Table 3.3).

Reason: The only possible directed edges in TRN, exist between tiers 1 → 2, 1 → 3, 2 → 2 and 2 → 3. The tier 3 nodes, which account for almost 90% nodes in the network, have no edges among them, explaining why TRNs have low graph density.

Table 3.3. Density of TRN graphs.

TRN type	*E. coli*	*S. cerevisiae*	human	Mouse
D	0.0015	0.00065	0.0010	0.0010

3.2.3. Small World Property. A small world network is one where it is possible to travel from one node to another in a limited number of hops [86]. Small world networks tend to possess a small diameter [1] [87].

Since information flows unidirectionally from tier 1 to 3 (as shown using the three tier topology), TRNs are weakly-connected graphs where every node is not reachable from every other node (i.e. undefined diameter). Thus, we use two metrics to demonstrate the small world property of TRN: (1) *diameter* of undirected TRN and (2) *average shortest path* from tier 1 to 3 nodes (defined below).

[1] Graph diameter ($\mathcal{D}$) is the greatest distance between any pair of vertices. It is calculated by finding the largest shortest path among all pair of vertices i.e., $\mathcal{D} = max_{u,v \in V} d(u, v)$, where $d(u, v)$ is the shortest path length between nodes u and v.

Given graph $G(V, E)$, $V = t_1 \cup t_2 \cup t_3$, tier 1 nodes $t_1 = \{u_1^1, u_1^2, \cdots\}$, tier 3 nodes $t_3 = \{u_3^1, u_3^2, \cdots\}$, average shortest path $< d >$ is defined as:

$$< d >= \frac{1}{|P|} \sum_{u_1^i} \sum_{u_3^i} d(u_1^i, u_3^i) \tag{3.2}$$

In the above equation P is the number of (u, v) node pairs such that $u \in t_1$, $v \in t_3$ and v is reachable from u.

Table 3.4. Diameter ($\mathcal{D}$) and average shortest path (<d>) of TRN.

TRN type	*E. coli*	*S. cerevisiae*	human	Mouse
$\mathcal{D}$	9	6	9	10
<d>	2.6	4.6	4.3	4.8

We intuit from the three-tier topology that the expected number of hops from a tier 1 to a tier 3 node should be 2 (tier 1 ⟶ tier 2, tier 1 ⟶ tier 3). Table 3.4 shows the diameter ($\mathcal{D}$) of the undirected TRN and its average shortest path (<d>). Note that the maximum diameter of the undirected TRN is 10. More importantly, the maximum average shortest path (Eq. 3.2) of TRN is 4.8, which is a direct demonstration of the small world property of a TRN [88, 89].

3.2.4. High Clustering Tendency.

We argue that the abundance of motifs in TRNs is a consequence of its tendency to form dense, tightly-knit groups, called *clusters*. The clustering tendency of any node u in an undirected graph $G(V, E)$ is measured in terms of its clustering coefficient, given by:

$$CC(G, u) = \begin{cases} 0, & \text{if } \delta(u) < 2 \\ \frac{2 \times t(u)}{\delta(u) \times (\delta(u) - 1)}, & \text{otherwise} \end{cases} \tag{3.3}$$

In Eq. 3.3, $t(u)$ is number of triangles node u participates in and $\delta(u)$ is its degree. The *average clustering coefficient* (ACC) of the undirected graph G is given by:

$$ACC(G) = \frac{1}{|V|} \sum_{u \in G} CC(G, u) \tag{3.4}$$

Eq. 3.3 elucidates that *ACC* is directly proportional to the number of triangles in an undirected graph. Table 3.5 shows that *ACC* of TRN is over 80 times that of E-R random graphs of same order and approximately same graph density.

Table 3.5. FFL count and average clustering coefficient.

	E. coli	R-*E.coli*	*Yeast*	R-*Yeast*	*human*	R-*human*	*Mouse*	R-*Mouse*
FFL	4798	18	4115	30	5850	23	2714	29
ACC	0.2110	0.0033	0.0830	0.0015	0.1200	0.0017	0.0970	0.0026

Relationship between motif abundance and clustering: This high ACC of TRN is commensurate with its motif abundance. The motifs (primarily the FFLs and bi fans) do not appear in isolation; they form dense clusters [90, 91, 92]. Investigation on the *E. coli* TRN topology indicate that there are 42 Feed Forward Loops (FFLs), that form six FFL motif clusters. (Appendix B Section 2 enumerates the FFLs across the three tiers in TRNs.) Similarly 208 bi fan motifs participate into two clusters. Table 3.5 shows that in addition to *ACC*, the number of FFL motifs in TRNs are significantly higher than their random counterparts.

Table 3.6. Percentage of positive (P), negative (N) and unknown (U) edges in TRN.

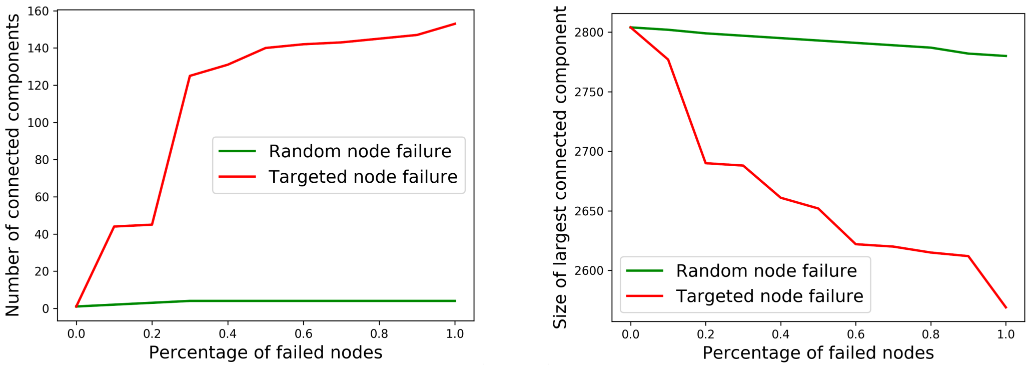
TRN type	*P*	*N*	*U*
E. coli	53.20	41.10	5.50
S. cerevisiae	–	–	–
human	33.51	20.45	46.03
Mouse	40.28	19.18	40.52

3.2.5. TF-Gene Regulation. We have discussed in Section 1.3 that TRNs are signed networks. The edges can be positive or negative, depending on the mode of interaction between the TF/TF or TF/ target gene. However, the information of the edge signs is not complete. Table 3.6 shows the number of positive, negative and unknown edge signs in each TRN. Note that the edge signs for *S. cerevisiae* TRN is not available.

3.2.6. Preferential Attachment Growth Model. We have discussed in Section 3.2.1 that TRNs have a scale free out-degree distribution. One approach to construct such networks is to employ a *preferential attachment growth model*, wherein when a new node is inducted into a network, it prefers to get attached to a node which has higher degree of connectivity [84].

As a consequence to preferential attachment, the hub nodes tend to acquire more and more links as the network grows. The implication of the existence of preferential attachment is that the hub node is the most preferred candidate of attachment for a new node. The probability of addition of an edge between a new node N and an existing node of u of degree k_u, $p(k)$, is either *linear* in the degree of node (i.e. $p(u) = \frac{k_u}{\sum_{v \in V} k_v}$) or it is *nonlinear* (i.e $p(u) = \frac{k_u^\gamma}{\sum_{v \in V} k_v^\gamma}$). There has been efforts to study the motif distribution of TRN with randomized networks generated by linear and nonlinear preferential attachment approaches [91, 93]. The preferential attachment-based topologies possessed FFLs that compared well in terms of abundance to the overall TRN of *E. coli*.

3.2.7. Robustness Against Random Node Failure and Vulnerability to Hub Node Failures. Robustness of a biological system is typically defined as the ability of the organism to retain its characteristic traits (called *phenotype*) in the face of genetic change (i.e. mutation) [94]. We are more interested to address robustness of TRNs from a graph theoretic standpoint. In our prior work, we define network *robustness* as the ability of the network to carry out information flow under node and link failures [95].



(a)

(b)

Figure 3.2. Random vs. targeted node failure. (a) Number of connected components in random vs. targeted node failures (b) Size of largest connected components in random vs. targeted node failures.

From our discussion in Section 3.2.1 we are aware that the TRNs exhibit scale free out-degree distribution. Scale free networks are inherently resilient to the failure of random nodes, yet vulnerable to the failure of hub nodes [96, 97]. The targeted failure or removal of the hub nodes in tiers 1 and 2 of the three tier topology is likely to knock off the majority of the poorly connected tier 3 nodes. Recall that approximately 90% nodes in TRN reside in tier 3. Nodes randomly picked for removal are most likely to belong to tier 3, which when removed, should not affect the overall connectivity of the TRN [79].

Taking cue from the known measures of network robustness [98, 99], we carry out a simple experiment wherein we knock off 0.1 – 1% (1) *randomly chosen nodes* in human TRN and (2) *targeted nodes* chosen with likelihood equal to their degree. Figure 3.2(a) and 3.2(b) shows that the targeted node failure results in the network fragmenting into significantly higher number of components as well as lower size of largest connected component as compared to its random counterpart.

3.3. INFERENCES

In this section we identify the significant topological attributes of Transcriptional Regulatory Networks (TRNs) such as *scale free out-degree distribution*, *low graph density*, *small world property*, *high clustering tendency* leading to the abundance of subgraphs called *motifs*, *TF-gene regulation*, *preferential attachment* and *robustness against random node failure*. In the subsequent sections, we shall see how each of these graph-theoretic properties can motivate the design of robust, energy-efficient networking solutions.

4. ROLE OF MOTIFS IN TOPOLOGICAL ROBUSTNESS AND INFORMATION FLOW

We have introduced the notion of motifs in Section 2.1. Let us now try to study their role in topological robustness and information flow within TRNs. From our earlier discussion, we have established that motifs are elementary circuits that may play vital role in robust information exchange. It has been shown that the bi fan motif renders dynamic stability in biological networks [100]. The robustness rendered by motifs in the event of topological or dynamical perturbation has been studied in [101]. The authors in [102] explain the role of positive feedback loops in TRN robustness, while [103] show that negative auto-regulation motif affects the mutational robustness of TRN. Our previous work attempted to exploit the inherent robustness of TRNs to design fault-tolerant wireless sensor network (WSN) topologies [45][104], focusing on the following fundamental similarities between a TRN and WSN: *First*, WSN nodes sample data from their sensing fields, just like genes exchange protein signals from the environment and neighbor genes. *Second*, WSN communication is critically impaired by node and link failures and interference, just as TRNs are affected by stochasticity in bio-chemical functioning of cells and alteration in molecular properties, leading to interference in signal exchange [37].

Apart from the above aspects, we are attempting to broach two other unanswered questions pertaining to FFL motifs: (a) their function in information flow in TRNs; and (b) their organization as building blocks leading to the formation of TRNs. We seek answers to both of these questions on the basis of certain networking yardsticks depicted in the flow diagram shown in Figure 4.1. Most of our experiments were carried out at a node level considering TRN nodes with high FFL motif participation, called *motif central nodes*. In this paper, we explore the purpose of FFLs in TRNs in the following order:

1. Role of FFLs in enabling *efficient communication* and *fault-tolerance* in TRNs.

2. The *topological structure* of TRNs with FFLs as building blocks.

3. The *functional role* of nodes with high FFL participation.

Let us now take a closer look at each of these directions. In Figure 4.1, we illustrate the summary of the different roles of FFLs in TRNs and the associated metrics that were used to analyze them.

- First, we define *communication efficiency* as a measure of how rapidly a node/motif can spread information to as many number of nodes in the network. As shown in Figure 4.1, we characterize *rapidity* of information spread through a widely used metric called network efficiency, defined as the average of the inverse of shortest path lengths [90, 105] and enumeration of number of paths created as a result of the FFL motif for details). Also, to capture the *spreading potential* of a node/motif we utilize the susceptible-infected-recovered (SIR) epidemic model. We also use centrality metrics such as closeness, betweenness and degree centralities, to corroborate our findings on information spreading potential of motif central nodes. Next, we define *fault-tolerance* as the ability of a network to continue communication despite component failure (such as a set of nodes) [106]. To quantify fault-tolerance, we gauge how the network efficiency and epidemic spread are affected when specific nodes are knocked off the network. Our earlier work on bio-inspired networking enabled wireless sensor networks to mimic a TRN topology for routing packets resulting in high communication efficiency and fault tolerance to random node/edge failures [107, 108, 109] leading to an indirect quantification of these TRN properties.

- Second, considering that FFL motifs can influence the information spread in TRNs, we analyze the *topological structure* of TRNs to identify their logical communication architecture comprising individual FFL units. As shown in Figure 4.1, we revisit two node-level metrics: (i) a characterization of TRN nodes into three tiers based on degree distribution (see Section 3) and (ii) FFL motif centrality of nodes. Based on these metrics, we find answers to questions such as connectivity, distance and among different classes (or roles *A*, *B* and *C*) of FFL motif central nodes, as well as their position in the three-tier hierarchy. Note that considerable effort has already gone into the analysis of the hierarchical structure of TRNs. Gerstein et al. studied the network interactions of different TFs and mRNAs in humans on the basis of properties such as hubs vs. non-hubs, connectivity, motifs, etc. [81]. Similarly, Bhardwaj et al. employed breadth-first search (BFS) to form a hierarchy of TFs based on regulating-regulated TF relationships to identify the master regulators in *E. coli* and *S.cerevisiae* TRNs [82]. Finally, the five-level hierarchy of TFs and operons in *E. coli* proposed by Ma et al. [29] is the closest to our proposed three-tier characterization. However, since the three-tier topology effectively combines the notions of motif centrality and degree centrality at a node level, it can help to (i) identify and differentiate between nodes with high degree (called *hubs*) and those with high FFL motif centrality on the basis of information spread and fault-tolerance (in Section 4.2) and (ii) explain the topological organization of TRN w.r.t. FFL motifs (in Section 4.2.3).

- Third, based on the role of FFL motif central nodes in information spread and topological organization of TRNs, we look for biological validation on the *functional properties* of motif central nodes in published literature. Figure 4.1 shows that we refer to the following three well-studied metrics to aid us in identifying these functional roles: motif clustering diversity, k-shell property and biological pathways. Motif clustering diversity (MCD) identifies the participation of a node in unique clusters of FFL motifs and has been proven to serve as a measure of TFs that serve as global regulators controlling the transcription of several genes. Another metric of importance is the participation of a motif central node in biological pathways, which may show their roles in signal transduction, metabolism or gene regulation.

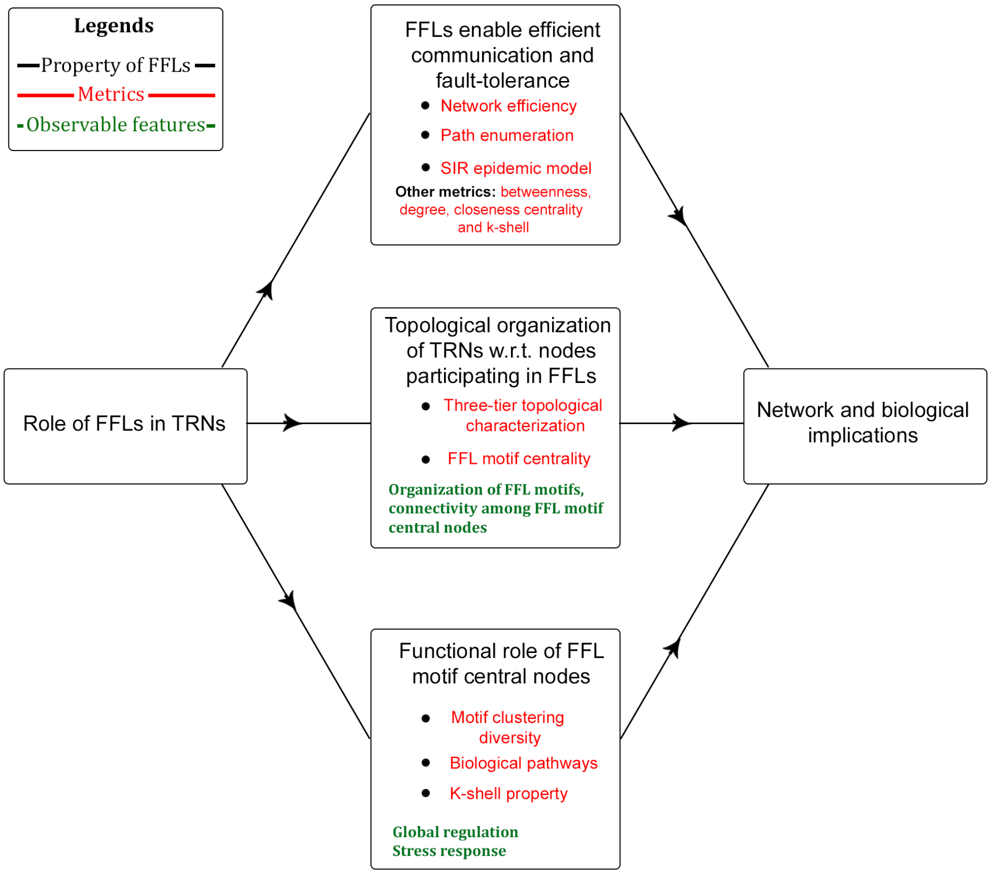

Legends
Property of FFLs
Metrics
Observable features
FFLs enable efficient communication and fault-tolerance
Network efficiency
Path enumeration
SIR epidemic model
Other metrics: betweenness, degree, closeness centrality and k-shell
Role of FFLs in TRNs
Topological organization of TRNs w.r.t. nodes participating in FFLs
Three-tier topological characterization
FFL motif centrality
Organization of FFL motifs, connectivity among FFL motif central nodes
Network and biological implications
Functional role of FFL motif central nodes
Motif clustering diversity
Biological pathways
K-shell property
Global regulation
Stress response

Figure 4.1. Goals and methodology of the proposed research.

Next, the k-shell value (or k-value) of a node quantifies whether it is located at the core (or periphery) of a network. Since nodes with high k-value were shown to be effective information spreaders, we study whether TFs with high k-value are also global regulators. We also validate our intuition that different classes of motif central nodes serve the function of regulators (from a communication efficiency angle) and cellular stress response (from a fault tolerance perspective).

Finally, we combine the three aforementioned aspects of FFLs and their roles in TRNs and draw some inferences on the *network and biological implications* of our findings. In network implications, we come up with a hub-and-spoke representation of the TRNs comprising motif central nodes that can lead us to new efficient network communication protocols. The purpose of biological implications is to briefly discuss the effects of different types of motif central nodes in the context of disease biology.

4.1. MOTIFS IN TOPOLOGICAL ROBUSTNESS

Let us consider a WSN graph $G_w(V_w, E_w)$ where the nodes V_w represent sensors, and edge $e(i, j) \in E_w$ exists if two nodes i and j are within transmission range of one another, and TRN graph denoted by $G_g(V_g, E_g)$. Following [45][104], we define a mapping function $M : G'_w \rightarrow G'_g$, where $G'_g(V'_g, E'_g)$ and $G'_w(V'_w, E'_w)$ are subgraphs of G_g and G_w respectively, such that $(u, v) \in E'_g$ exists if and only if there exists a path between $M(u)$ and $M(v)$ in G'_g. Here G'_g is the *mapped-TRN subgraph* and the corresponding G'_w is termed *bio-WSN*. While this embedding approach does not provide an exact mapping between all bio-WSN and TRN subgraph nodes, bio-WSNs have already been shown to preserve the node connectivity and motif abundance properties of TRN subgraphs. Therefore, the information flow in bio-WSN graphs may be considered to be an effective representation of signal propagation in TRNs. Simulations have shown that bio-WSNs exhibit significant improvement in packet delivery rate, network latency and lifetime over Erdös Rényi random graph-based WSNs, even under node and link failures. However, a major limitation to our previous work is that it fails to explain which specific graph attribute of TRN is responsible for such improvement.

We believe that knowledge of the specific graph property that lends topological robustness to TRNs could be extended to design other robust networks that mimic the robustness of TRNs. This work is the first step in that direction, where we graph-theoretically explore the role played by motifs in information flow in TRNs. In particular, we (i) analyze why a 3-node subgraph called Feed Forward Loop (FFL) typifies robust signal propagation via TRN motifs, (ii) utilize centrality metrics to show how motifs affect the topological robustness and resilience of TRN and corresponding bio-WSN subgraphs and (iii) validate our findings through graph-theoretic and simulation experiments. In our study, we define *robustness* as the ability of a network to carry out information flow under node and link failures, and *resilience* as its ability to preserve shortest path lengths despite such failures.

4.1.1. Known Functions of Motifs. Here we will first introduce some of the known functional and topological roles of motifs in a TRN.

Motifs as signal propagating circuits: Motifs are elementary circuits for signal propagation that aid in gene transcription leading to protein synthesis [110]. They play the role of filters, pulse generators, response accelerators and temporal pattern generators.

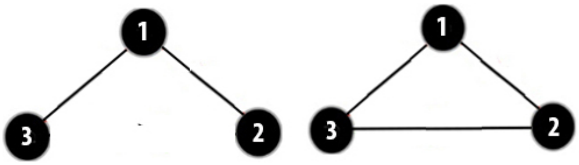


Figure 4.2. Triplets. (a) Open Triplet and (b) Closed Triplet.

Clustering tendency of TRNs: We define a *connected triplet* as three nodes that are connected by either two undirected edges (open triplet) (Figure 4.2 (a)) or three undirected edges (closed triplet) (Figure 4.2 (b)). The *average clustering coefficient (ACC)* measures the clustering tendency of an undirected graph, and is calculated as the ratio of the number of closed triplets to the total number of connected triplets.

Table 4.1. Comparison of ACC of undirected TRN and Erdös Rényi Random Graphs.

Graph	$Random_E$	*E. coli* TRN	$Random_Y$	Yeast TRN
Triangles	11	1405	26	3750
ACC	0.0015	0.211	0.001	0.083

Clearly, nodes with high clustering tendency belong to well-connected neighborhoods. In *E. coli* and Yeast TRNs we have two directed triangles: cyclic triangle called *Feedback Loop (FBL)* and the more frequent acyclic triangle called *Feed Forward Loop*

(FFL), as illustrated in Figure 4.4 (a). Table 4.1 shows that undirected *E. coli* and Yeast TRN graphs possess significantly more triangular motifs than Erdös Rényi random graphs of similar sparseness as *E. coli* (Random$_E$) and Yeast (Random$_Y$), thus exhibit higher ACC.

Communication Pathway Alternatives: Motif structures often possess multiple paths between a pair of nodes. For instance in the motif structure S_1 illustrated in Figure 4.4 (b), there are three paths between nodes 1 and 4, namely $p_1 = \{1, 2, 4\}$, $p_2 = \{1, 3, 4\}$ and $p_3 = \{1, 2, 3, 4\}$. Therefore, the multiplicity of such paths among node pairs should allow TRN graph to remain connected despite failures of nodes and links.

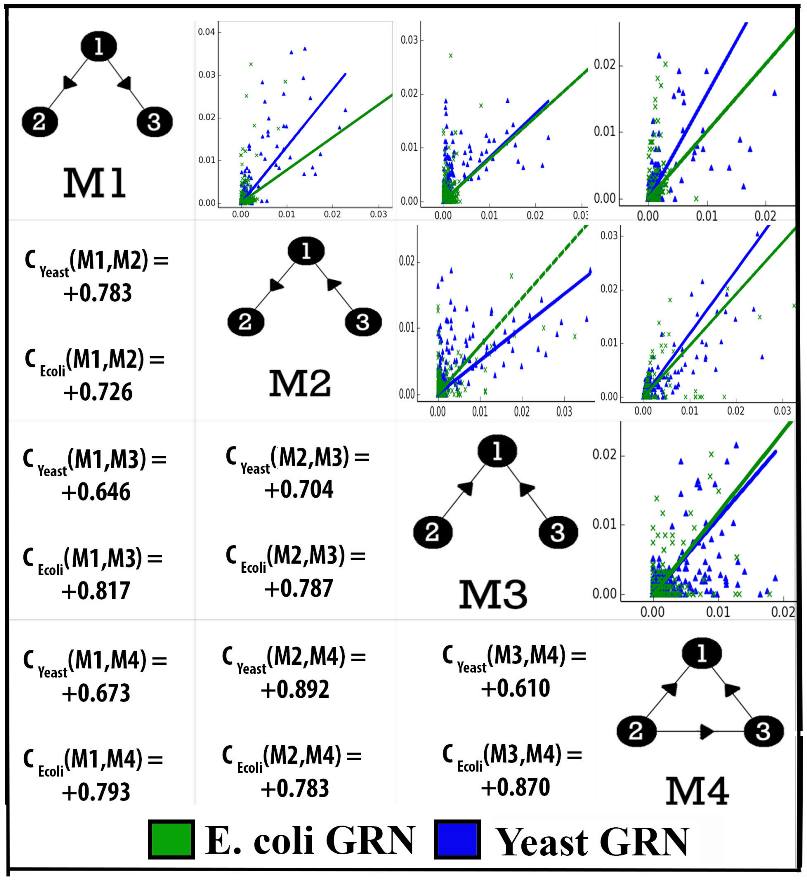


Figure 4.3. Inter-motif correlation plot. A point (x, y) exists in the scatter plot in row i and column j, if a node participates in x M_i motifs and y M_j motifs.

4.1.2. Motif and Motif Central Nodes. Based on two criteria, we select the simplest subgraph, i.e. the Feed Forward Loop (FFL), that typifies the role of motifs in topological robustness of TRN. FFLs are also the building blocks of larger motifs in TRNs (as shown in Appendix A Section 2).

1. The motif must be statistically significant so that its properties are applicable to TRN and its subgraphs.
2. It should participate in multiple paths among node pairs since multiplicity of pathways influences network connectivity during node or link failures.

Existence of motif central nodes: We estimate the three most frequent 3-node motifs (labeled $M1$, $M2$ and $M3$ in Figure 4.3), using the network motif detection tool called *FANMOD* [24]. $M1$, $M2$ and $M3$ are all subgraphs of another 3-node motif FFL (labeled $M4$) in Figure 4.3. To understand whether there exist a set of nodes that participates in high number of motifs $M1$ to $M4$, we plot a 4×4 inter-motif correlation graph (shown in Figure 4.3). A point (x, y) exists in scatter plot in row i and column j, if there is a TRN node participating in x M_i motifs and y M_j motifs. High inter-motif correlation coefficient (lower triangle of Figure 4.3), especially among dissimilar motifs $M1$, $M2$ and $M3$, confirm the existence of a few nodes, called *motif central nodes*, that participate in many $M1$ to $M4$ motifs and are likely to control the information flow in TRN.

Feed Forward Loop (FFL) as chosen motif: Let us consider the insights for the choice of FFL motifs in our study. First, we have discussed in Section 4.1.1 that triangular motifs are responsible for the high clustering tendency of a graph, and also, nodes with high clustering coefficient belong to well-connected neighborhoods. FFLs, being the most abundant triangular TRN motifs, are likely to participate in multiple paths (between other node pairs). Second, FFL is the most elementary TRN motif to have two paths from node S to T: one direct path (shown in green in Figure 4.4 (a)) and the other via I (shown in red), again alluding to the fact that such cascades of FFLs participate in high number of paths.

Taking these points into consideration, we take FFL as our chosen motif. As shown in Appendix A Section 2, frequent 4, 5 and 6-node motifs in *E. coli* and Yeast have FFL as building blocks.

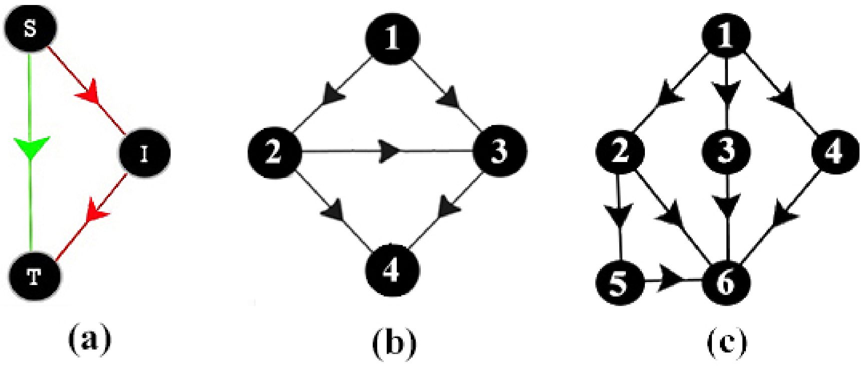


Figure 4.4. Subgraphs to analyze FFL based centrality. (a) FFL (b) Subgraph S_1 (c) Subgraph S_2.

4.1.3. Evaluation Metrics. In this section we introduce some centrality metrics in order to study the role of FFLs in robust information flow. Section 4.1.3.7 elaborates on our intuition behind these metrics in analyzing robustness and resilience rendered by motifs. We explain the metrics using subgraphs S_1, S_2 (Figures 4.4 (b) and (c)). In rest of the paper, we use the term motif interchangeably with FFL. These metrics are applicable to any directed graph $G(V, E)$ where $e(i, j) \in E$ is a directed edge from node i to node j.

4.1.3.1. Node motif based centrality (NMC). Here we introduce the metric used to gauge the participation of a node in FFL motif.

4.1.3.1.1. Directed Triplet:. For each combination of $i, j, k \in V$, directed triplet is calculated as:

$$\delta(i,j,k) = \begin{cases} 1 & if \quad e(i,j), e(j,k), e(i,k) \in E \\ 0, & \text{otherwise} \end{cases}$$

Since both *E. coli* and Yeast TRNs are directed graphs, $\delta(i,j,k)$, $\delta(j,i,k)$ and $\delta(j,k,i)$ are not equal.

4.1.3.1.2. Node Motif Centrality (NMC). is the number of motifs a node participates in. Given that FFL is the motif of our study, NMC of node i is calculated as follows:

$$NMC(i) = \sum_{j,k \in V} \delta(i,j,k) + \delta(j,i,k) + \delta(j,k,i) \tag{4.1}$$

In graph S_1 in Figure 4.4(b), $NMC(1) = 1$ and $NMC(2) = 2$. We define *Motif central nodes (C)* as a set of nodes with NMC greater than median NMC of the graph G.

4.1.3.2. Path and node betweenness centrality. *Simple path* is one in which no node is visited more than once. Two paths between a node pair are called *independent* if they contain no common nodes except source and destination nodes. In graph S_1 in Figure 4.4(b), $p_1 = \{1,2,4\}$ and $p_2 = \{1,3,4\}$ are independent paths from node 1 to 4.

For each node pair $i, j \in V$, we define a family of sets, called *independent path groups* $I(i,j)$, such that each member, denoted by $I_l(i,j)$ $(0 < l \leq |I(i,j)|)$, is a *maximal* set of mutually independent paths between i and j. For any l, m, where $0 < l, m \leq |I(i,j)|, l \neq m$, there exists at least one path in $I_l(i,j)$ that is not independent to at least one path $I_m(i,j)$. For instance, for node pairs 1 and 6 in graph S_2 (Figure 4.4(c)), we consider four paths: $p_1 = \{1,2,6\}$, $p_2 = \{1,3,6\}$, $p_3 = \{1,4,6\}$ and $p_4 = \{1,2,5,6\}$. These paths can be assigned to two member sets of independent path group $I(1,6) = \{I_1(1,6), I_2(1,6)\}$: $I_1(1,6) = \{p_1, p_2, p_3\}$ and $I_2(1,6) = \{p_2, p_3, p_4\}$ such that $p_1 \in I_1(1,6)$ and $p_4 \in I_2(1,6)$ are not independent. Evidently, any given path may belong to several member sets of an independent path group. In the above example, paths p_2 and p_3 belong to both member sets $I_1(1,6)$ and $I_2(1,6)$ in $I(1,6)$.

4.1.3.3. Path centrality (PC). is the number of simple paths between all pair of nodes that a given node intercepts. Given that $p(i,k|j)$ is number of simple paths between nodes i and k that pass through node j, $PC(j) = \sum_{i,k \in V} p(i,k|j)$.

4.1.3.4. Independent path centrality (IPC). is number of independent paths between all node pairs that a node intercepts. We define $IPC(j) = \sum_{i,k \in V} \sum_{l} |I_l(i,k|j)|$. Here $|I_l(i,j|i)| = 0$ and $|I_l(i,j|j)| = 0$, and $|I_l(i,k|j)|$ is number of times a node j occurs in a member path of set $I_l(i,k|j)$, such that node j is neither source nor target in member paths.

4.1.3.5. Node betweenness centrality (NBC). is the fraction of total shortest paths between any node pairs that pass through a given node. For any $i, k \in V$, let $\sigma(i,k)$ be the number of shortest (i,k)-paths and $\sigma(i,k|n)$ is the number of those (i,k)-paths passing through node n, then we define $NBC(n) = \frac{\sigma(i,k|n)}{\sigma(i,k)}$.

Betweenness Index (BI(P)) is the ratio of the sum total NBC of a set of nodes P to sum total NBC of all the nodes in a graph. $BI(P)$ can be written as follows:

$$BI(P) = \frac{\sum_{i \in P} NBC(i)}{\sum_{j \in V} NBC(j)} \tag{4.2}$$

4.1.3.6. Network efficiency and data forwarding index. Let us briefly discuss the metrics that help gauge data forwarding in a network.

4.1.3.6.1. Network Efficiency. measures the average shortest path. For any graph G, efficiency $\eta(G)$ is:

$$\eta(G) = \frac{1}{T} \sum_{i,j \in V} \frac{1}{d(i,j)} \tag{4.3}$$

where, T = Number of existing paths among all node pairs and $d(i,j)$ = Shortest path length from node i to node j

Since efficiency considers reciprocal shortest path length between node pairs, higher $\eta(G)$ implies lower average shortest path. Let $nf\%$ nodes be randomly removed from graph G resulting in a graph G_{nf}. We calculate *percentage drop in efficiency* due to (increase in average shortest path length on) failure of $nf\%$ nodes as $U_{\eta}(G_{nf}) = 100 \times (1 - \frac{\eta(G_{nf})}{\eta(G)})$. Analogously, we calculate percentage drop in efficiency due to failure of $ef\%$ edges $U_{\eta}(G_{ef})$ using similar formulation.

4.1.3.6.2. Data Forwarding Index (DI(P)). , similar to BI(P), is the ratio of the sum of data packets forwarded by a set of nodes P to sum of packets forwarded by all nodes in a graph. For any graph, $DI(P)$ can be written as follows:

$$DI(P) = \frac{\sum_{i \in P} \mathit{No.\ of\ Packets\ Forwarded\ by\ i}}{\sum_{j \in V} \mathit{No.\ of\ Packets\ Forwarded\ by\ j}} \tag{4.4}$$

Both $BI(P)$ and $DI(P)$ attempt to gauge the collective participation of a set of nodes P in the shortest paths and information flow among other node pairs in G.

4.1.3.7. Robustness and resilience. Newman et al. have defined network robustness in terms of the number of independent paths between node pairs [111]. This notion of independent paths is in keeping with *Menger's theorem of Vertex Connectivity*, which states that minimum number of vertices whose removal disconnects two nodes is equal to the maximum number of pairwise vertex-independent paths [112]. We see in Figure 4.4(b) that there are two independent paths between nodes 1 and 4, and therefore, at least two nodes (2 and 3) must be removed to disconnect 1 and 4.

Let us discuss how the three centrality metrics, defined in Section 4.1.3, can help assess the *robustness* rendered by a node. First, if a node has high PC and IPC indices, it is highly likely to participate in network traffic flow. Also, from Menger's theorem on vertex connectivity, we infer that nodes with high IPC offer more topological robustness against failures. Second, nodes with higher NBC intercept many pairwise shortest paths, again facilitating information flow with minimum delay.

With regard to *resilience*, removal of a certain fraction of nodes or links from any graph is likely to result in the increase in shortest path length between other node pair. As pointed out in Section 4.1.3, an increase in average shortest path length of a graph would result in a drop in network efficiency. We define graph resilience as its ability to preserve *network efficiency* when a certain fraction of nodes or links fails. We intuit that w.r.t. each FFL motif, *the failure of the direct link between source S and target T causes the shortest path length between S and T to increase only by a single hop (Figure 4.4(a))*.

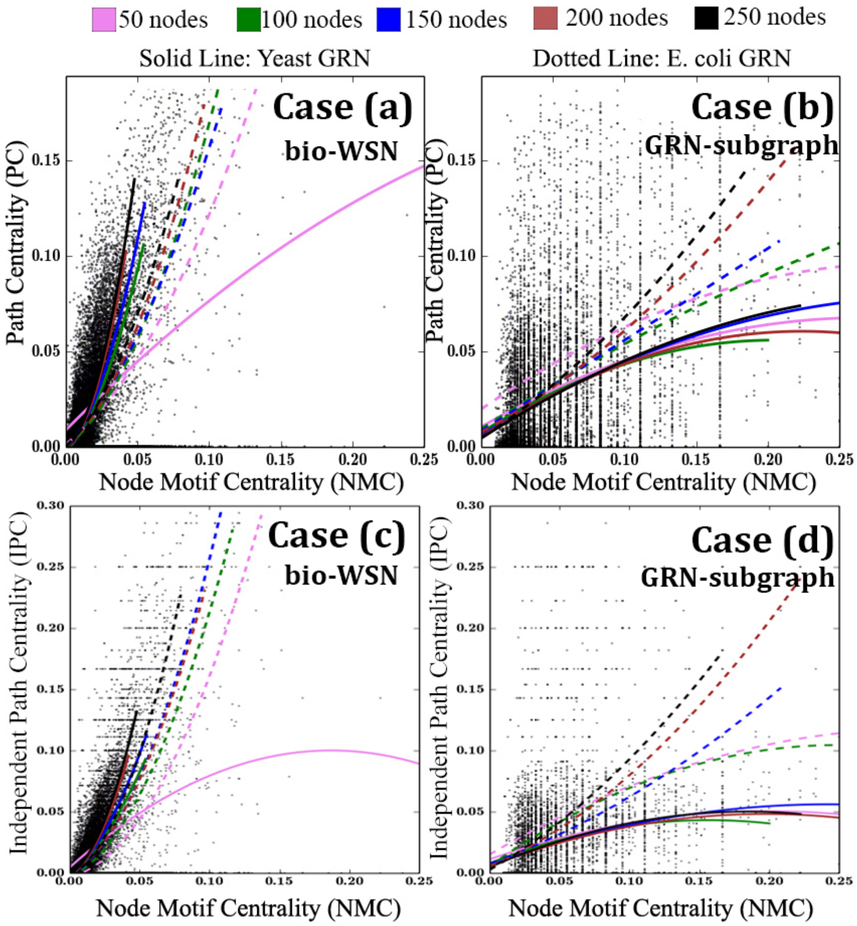


Figure 4.5. Centrality correlation. Cases (a) and (b): NMC vs. PC and IPC for *E. coli* Cases (c) and (d): Yeast bio-WSN and TRN subgraphs.

4.1.4. Experimental Results. In this section we discuss the results from graph-theoretic and simulation experiments, performed on 50 *E. coli* and Yeast TRN subgraphs (acquired using GeneNetWeaver [11]) and corresponding bio-WSNs, of sizes 50, 100, 150, 200 and 250 nodes. PC, IPC and NMC are calculated using the Python NetworkX Library [113]. The number of packets forwarded, motif participation, path or independent path participation for any node depend on the size and density of a graph. Thus in our experiments, we normalize each metric (PC, IPC, NMC or number of packets forwarded) for each node, by the aggregate sum of that given metric for the entire graph.

4.1.4.1. Graph-theoretic analysis. In the first two experiments we seek first-hand topological evidence of the functional role of motifs in forming robust pathways of signal propagation. In Section 4.1.3.7, we have discussed reasons why PC and IPC are effective metrics for robustness rendered by any node. Here we plot PC and IPC against NMC, for each node in Yeast and *E. coli* bio-WSNs and TRN subgraphs. We then apply nonlinear regression to obtain best fit lines from the scatter plot.

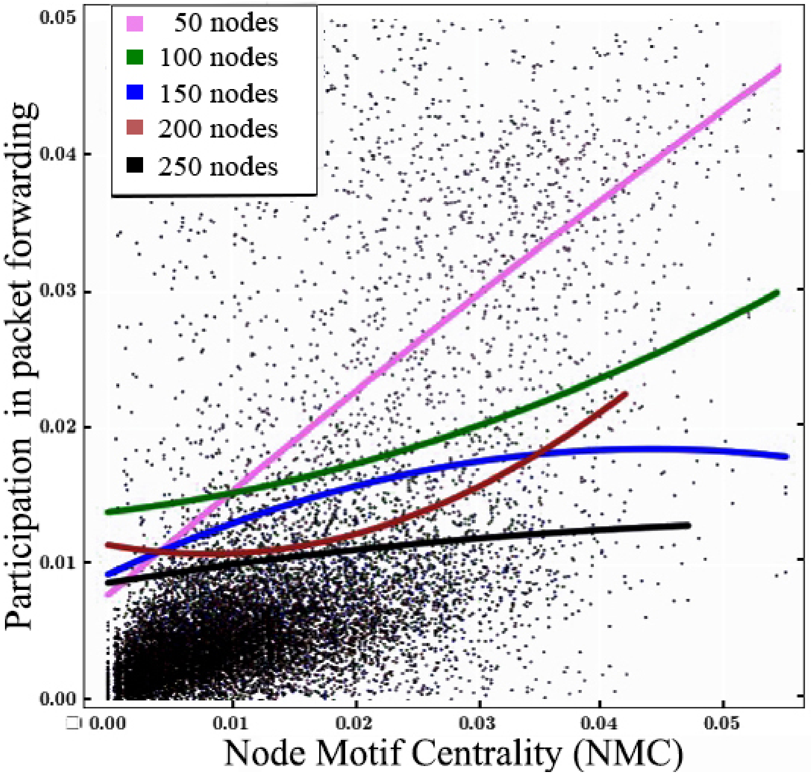


Figure 4.6. NMC vs. Packet Forwarding in Yeast bio-WSNs.

Plots in Figure 4.5 shows that for both, bio-WSN and TRN subgraphs, PC (Cases (a) and (b)) and IPC (Cases (c) and (d)) increases with NMC, showing that motifs indeed form pathways for information flow. It is noteworthy that in all the cases, regression lines grow steeper for larger bio-WSN and TRN subgraphs. Therefore, we infer that participation of motif central nodes in information flow grows with graph size.

4.1.4.2. WSN simulations. For each of the 250 Yeast bio-WSNs, we run simulations on OMNET++ Castalia [114] for a duration of 10 minutes. We use Collection Tree Protocol (CTP), which is a distance vector routing protocol used for WSN communication [115]. Here 5% of the nodes with highest in-degree are selected as sink nodes.

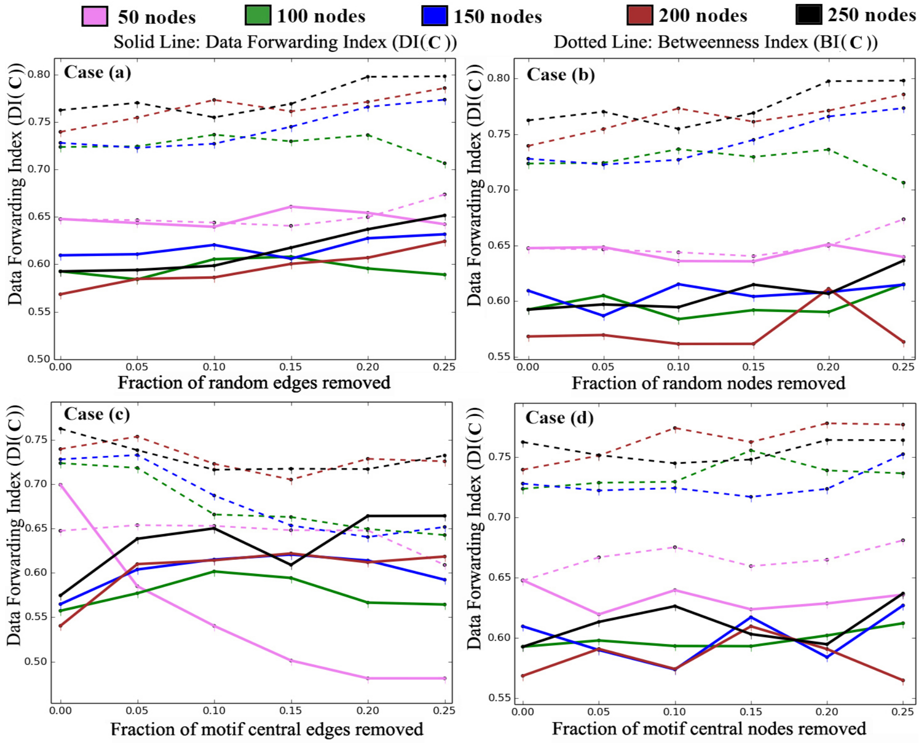


Figure 4.7. Participation of set of motif central nodes in shortest paths (dotted line) and packet forwarding (solid line) in Yeast bio-WSNs during 4 cases. (a) Random edge failure (b) Random node failure (c) Motif central edge failure (d) Motif central node failure.

4.1.4.2.1. Packet forwarding of motif central nodes. Given that bio-WSN graphs preserve the node connectivity of TRN, information flow via packet forwarding in bio-WSN graphs is quite analogous to the protein signal propagation in TRN. Here we analyze whether the NMC of a node has any bearing on its packet forwarding. Figure 4.6 shows that the increase in NMC causes an overall increase in packet forwarding index. We conclude that high packet forwarding of motif central nodes is a consequence of their high PC indices.

4.1.4.2.2. Packet forwarding of motifs during node or link failure. This experiment explores whether the motif central nodes (C) continue to provide pathways for signal propagation in four failure cases: (a) random edge failure, (b) random node failure, (c) motif central edge failure and (d) motif central node failure. *Motif central edges or links*, similar to motif central nodes, are defined as a set of edges with motif centrality greater than median motif centrality of all edges in a graph. In each case, we calculate the average BI and DI of C in Yeast bio-WSNs according to Eq. (4.2) and (4.4). The plots in Figure 4.7 indicate that motif central nodes consistently participate in over 70% shortest paths and $50 - 65\%$ of packet forwarding, for all failure cases of random and targeted nodes and links.

4.1.4.2.3. Resilience rendered by motifs. In our previous experiments we have seen that motif central nodes participate in bulk of information flow in TRNs and bio-WSNs during myriad failure conditions. We now revisit our insight regarding structure of FFL motif: failure of direct link from source (S) to sink node (T) offers an indirect path which is only a single hop longer. *We intuit that high frequency of FFLs should lead to little increase in average shortest path due to failures and consequent preservation of network resilience*. This experiment on resilience is therefore an extension of Case (a) and Case (b) in experiment 2 of Section 4.1.4.2. We plot the percentage drop in efficiency (U_η) and percentage drop in motif count, due to failure of random nodes and random edges in Yeast TRNs, bio-WSNs and Erdös Rényi random graphs.

We observe that in most cases, random graphs, characterized by notably fewer motifs than TRNs (as shown in Table I), undergo maximum drop in efficiency (depicted with dotted lines in Figures 4.8 Cases (a) and (b)). Combining results of Sections 4.1.4.2.2.

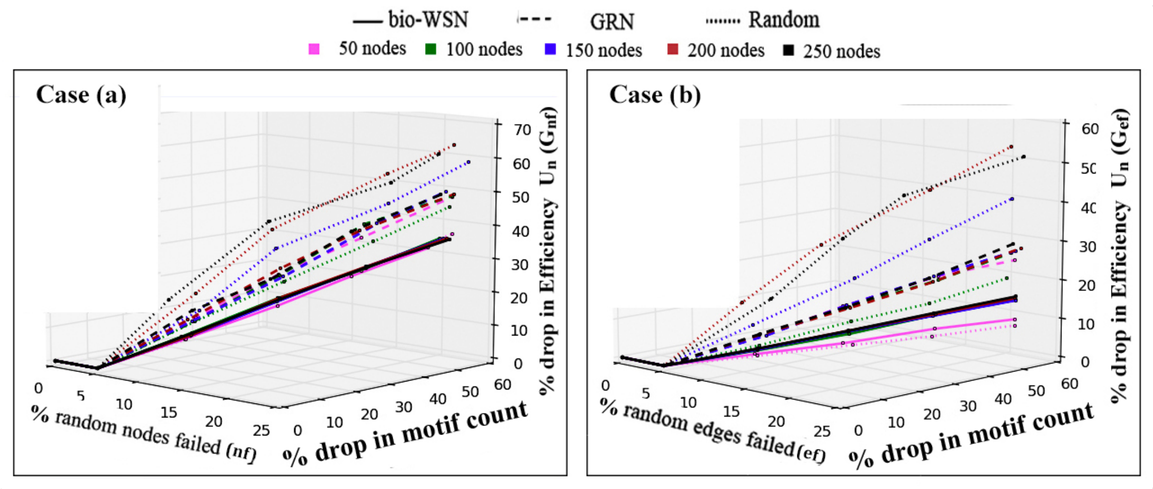


Figure 4.8. Efficiency and motif drop under node failure. Under failure of (a) $nf\%$ random nodes (b) $ef\%$ random edges.

and 4.1.4.2.3., we infer that motifs participate in communication paths in TRN and bio-WSNs. They also preserve network resilience by minimizing increase in average shortest path lengths under node or link failures.

4.2. COMMUNICATION EFFICIENCY AND FAULT-TOLERANCE

In this section, we explore other facets of FFL motifs from the standpoint of communication efficiency and fault-tolerance. First, we observe how the network efficiency is affected when the role A and B motif central nodes are knocked off from the TRN. Second, we use the SIR epidemic model to demonstrate the information spreading potential of role A and B motif central nodes. It is interesting to observe how the results differ when the same experiments are repeated with the selection of random nodes as well as nodes with high out-degree (i.e. hubs). We carry out five types of node selection:

- Random node: Select randomly from the node set of each TRN topology.

- Role A motif central node: Select nodes based on a likelihood proportional to their role A FFL motif centrality. In any TRN topology G, the probability of selection of node u with role A motif centrality $\delta_A(u)$ is given by $\frac{\delta_A(u)}{\sum_{v \in V} \delta_A(v)}$.

- Role B motif central node: Select nodes based on a likelihood proportional to their role B FFL motif centrality. The probability of selection of node u with role B motif centrality $\delta_B(u)$ is given by $\frac{\delta_B(u)}{\sum_{v \in V} \delta_B(v)}$.

- Total motif central node: Select nodes based on a likelihood proportional to their total FFL motif centrality (i.e. sum total of roles A, B and C). Probability of selection of node u with motif centrality $\delta(u)$ is given by $\frac{\delta(u)}{\sum_{v \in V} \delta(v)}$.

- Hub node with low role A motif centrality: Select nodes with high out-degree and low role A motif centrality, i.e. with likelihood proportional to ratio of node out-degree to role A motif centrality. Thus, in any TRN topology G, the probability of selection of node u with role A motif centrality $\delta_A(u)$ and out-degree $d_O(u)$ is given by $\frac{d_O(u)/\delta_A(u)}{\sum_{v \in V} d_O(v)/\delta_A(v)}$.

We remove $0.1\%, 0.2\%, \cdots, 0.5\%$, nodes from the input TRN graph using each of our node selection strategies and measure how the network efficiency changes in the event of node failures. Figure 4.9(a) shows that failure of random nodes and hub nodes of low role A motif centrality cause the least drop in network efficiency, followed by role B motif central node and total motif central nodes, in human TRN. The failure of role A motif central nodes cause the maximum drop in network efficiency.

Figures 4.9(b) and 4.9(c) show the plots for the mean evolution of infected nodes in human TRN for $\frac{\beta}{\gamma} < 1$ and $\frac{\beta}{\gamma} > 1$. Information flows the fastest for role A motif central nodes, followed by total and role B motif central nodes. Random node failure and hub nodes with low role A centrality have the least spread. Thus, role A motif central nodes are better information spreaders than hub nodes with low role A motif centrality in a TRN particularly when $\frac{\beta}{\gamma} > 1$, showing that the role A nodes retain infection for longer duration.

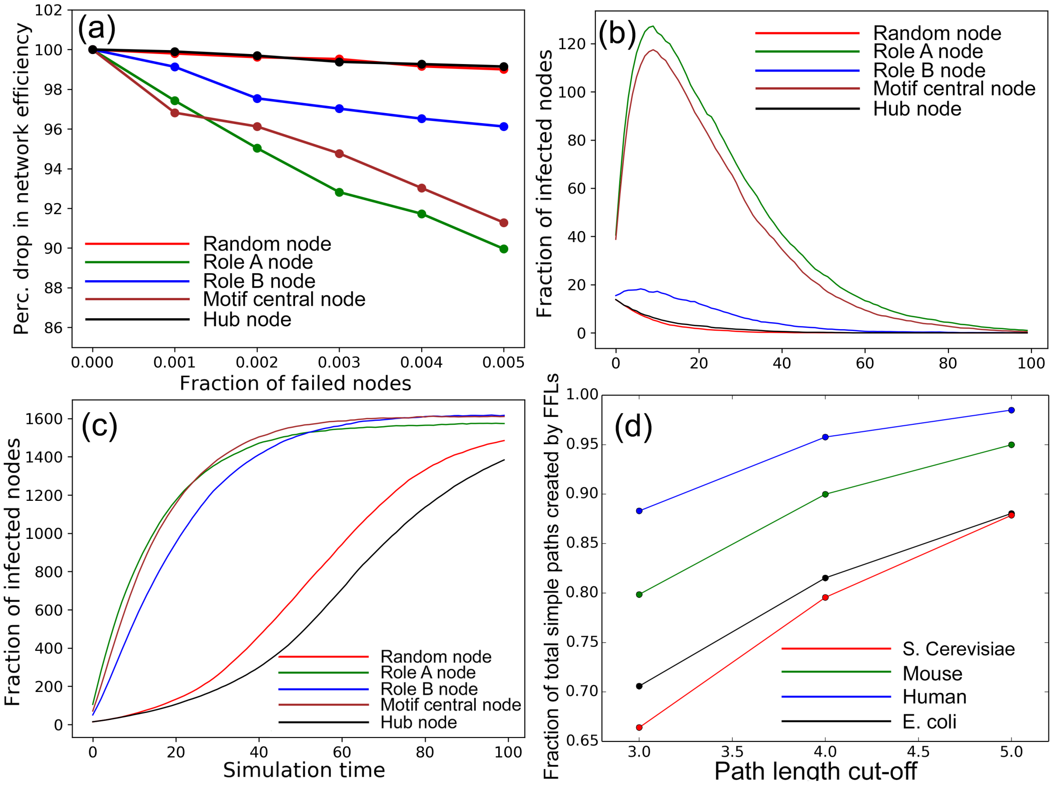


Figure 4.9. Motif central nodes in information spread. (a) Percentage of network efficiency during node failure in human TRN; Infection propagation using SIR model in human TRN for (b) $\frac{\beta}{\gamma} < 1$ $(\beta = 0.02, \gamma = 0.1)$ (c) $\frac{\beta}{\gamma} > 1$ $(\beta = 0.1, \gamma = 0.02)$ (d) Fraction of total simple paths created in TRN by FFL motifs

4.2.1. Motif Central Nodes: Good Spreaders of Information. Let us discuss potential reasons from a graph-theoretic standpoint that make motifs good information spreaders in TRNs.

4.2.1.1. Centrality. We next evaluate the correlation between role A motif central nodes and the graph centrality metrics such as *degree*, *closeness* and *betweenness*. For each correlation, we find the scatter plot and apply nonlinear regression to obtain best fit lines. The plots show that there is a moderate to strong correlation between normalized role A motif centrality (NMC) and normalized betweenness, degree and closeness centrality values corroborating that role A motif central nodes are good information spreaders.

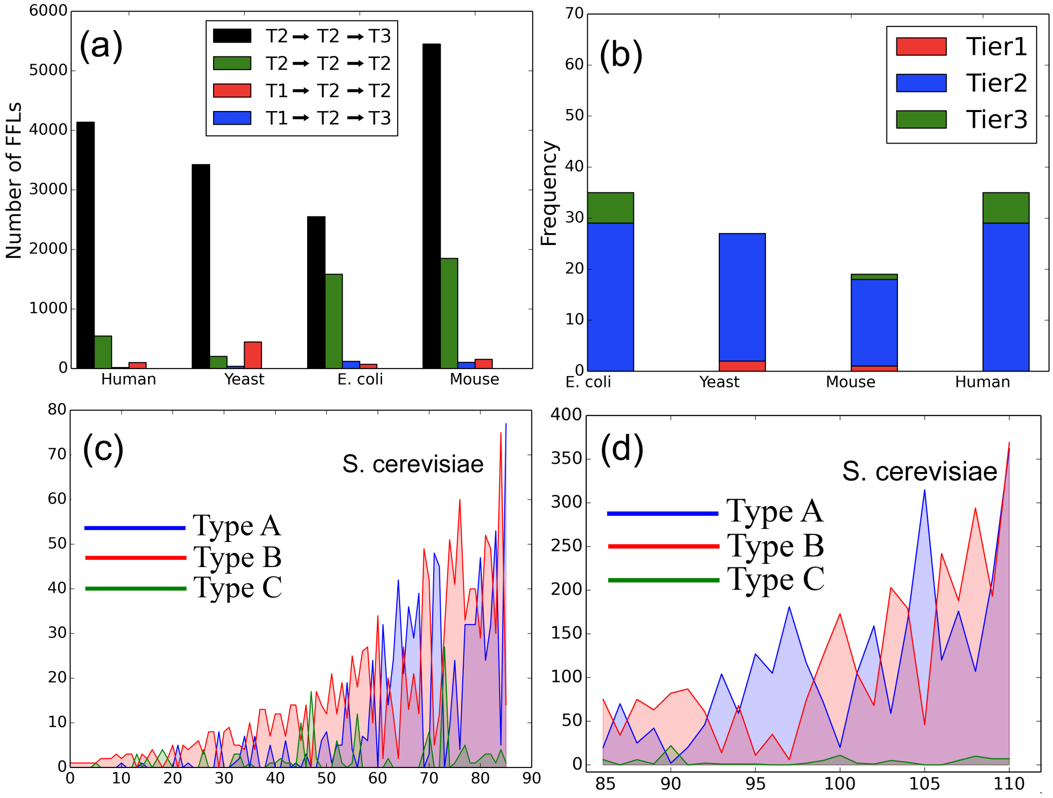


Figure 4.10. Interplay between FFL motif centrality and tiers. (a) Distribution of FFL motifs across the three tiers in TRNs (b) Classification of high motif central nodes across tiers; role A, B and C participation; X axis: Nodes in tier 2 arranged in the non-decreasing order of motif centrality; Y-axis: Frequency of role A, role B and role C motif centrality (c) FFL motif centrality $<$ 100 and (d) FFL motif centrality $>$ 100

4.2.1.2. K-shell decomposition. Kitsak et al. showed that the most efficient information spreaders are located in the inner core of the network (i.e. high $k-$value as defined in Appendix A Section 4), fairly independently of their degree [116]. The relatively high correlation between role A motif centrality and k-shell property evidences that the role A motif central nodes belong in highly-connected neighborhoods in TRNs, making them rapid information spreaders.

4.2.2. Participation of FFL Motifs in Simple Paths in a TRN. We apply our proposed heuristic (see Appendix A Section 3) with maximum considered path-length $pLimit = 3, 4, 5$ on TRNs of *E. coli*, *S.cerevisiae*, human and mouse TRN topologies. Our results show that the direct and indirect links of FFL motifs are responsible for creating a majority of the paths in TRNs (Figure 4.9(d)).

4.2.3. Topological Organization of TRN w.r.t FFLs. We are interested in studying the nodes of a TRN based on the types of motif centrality (role A, B and C) and also their topological organization in the three tier TRN architecture. *We define high motif central nodes as nodes with FFL motif centrality greater than or equal to* 100*; although this cut-off is arbitrary, it roughly accounted for* $\sim 1 - 2\%$ *of the TRN nodes.* In this section, we use the abbreviation NMC to denote total node motif centrality (δ).

4.2.3.1. Distribution of FFL motifs across tiers. Considering the direction of edges across tiers, we infer that FFL motifs can exist in the forms (a) $T1 \rightarrow T2 \rightarrow T2, (b)T1 \rightarrow T2 \rightarrow T3$, (c) $T2 \rightarrow T2 \rightarrow T2$ and (d) $T2 \rightarrow T2 \rightarrow T3$. Figure 4.10(a) depicts that majority of FFL motifs exist among $T2 \rightarrow T2 \rightarrow T3$ and $T2 \rightarrow T2 \rightarrow T2$.

4.2.3.2. Most high NMC nodes belong to tier 2. In Figure 4.10(b), we classify the high motif central nodes across tiers to show that maximum number of FFL motif central nodes belong in tier 2.

4.2.3.3. Most nodes in tier 2 have both role A and B properties. The TRN nodes (excluding those with non-zero FFL motif centrality) are ranked in increasing order of FFL motif centrality. The role A, B and C participation is calculated for two cases: (i) FFL motif centrality < 100 and (ii) ≥ 100 in TRN. In Figures 4.10(c) and 4.10(d), we show that, for both cases, tier 2 nodes predominantly possess role A and B properties.

4.2.3.4. Average distance between motif central nodes. The average shortest path (between all pair of nodes having motif centralities ≤ 25) tends to decrease as the motif centrality value increases, suggesting that the higher motif central nodes are closer to one another in the TRN topology (heat map in Figure 4.11(a).

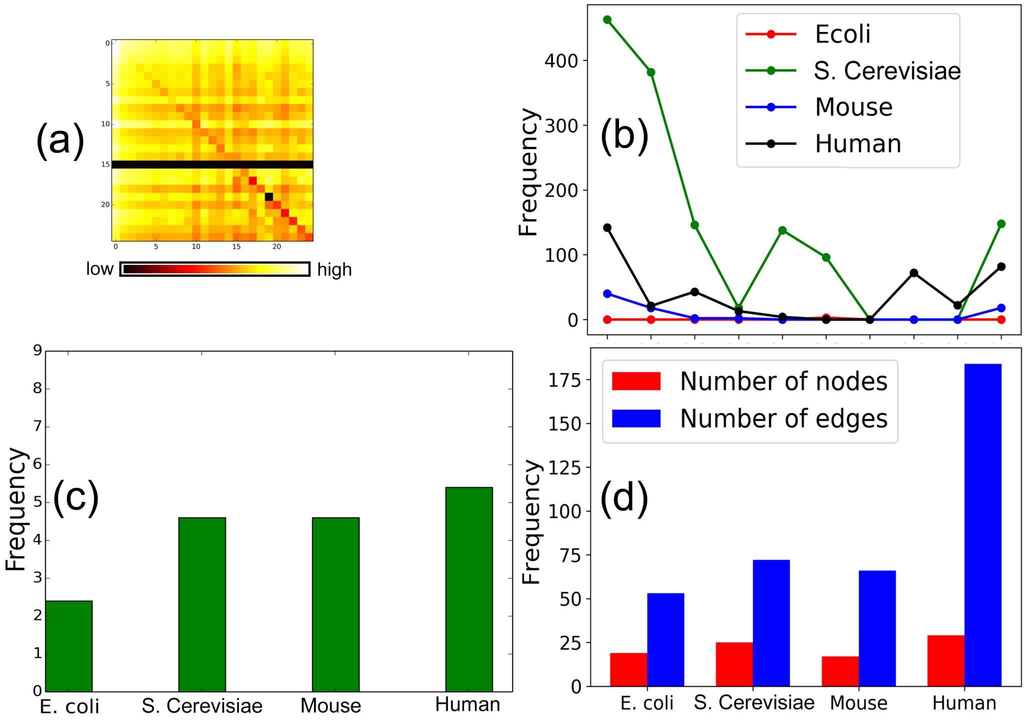


Figure 4.11. Relationship among motif central nodes. (a) Average distance between motif central nodes in human TRN (b) Intermediary nodes connecting high FFL motif central nodes (c) Average participation of high motif central nodes in tier 2 in each others' motif clusters (d) Large number of FFL motif central nodes are directly connected

4.2.3.5. High NMC nodes are connected by low NMC nodes as intermediaries.

We next generate 10 discrete levels $(0.1, 0.2, \cdots, 1.0)$ of FFL motif centrality with respect to the maximum FFL motif centrality in the topology. For instance, if a node belongs to level 1.0, it implies NMC is between $90 - 100\%$ of the maximum NMC in the TRN. Figure 4.11(b) shows that the low motif central nodes (belonging to level 0.1 and 0.2) serve as intermediary nodes connecting high node motif central nodes belonging in tier 2, although a notable number of intermediate nodes are high NMC nodes (i.e. belong to level 1.0).

4.2.3.6. High motif central nodes belong in each others' clusters. Gorochowski et al. [117] showed that FFL motifs in complex networks exist in clusters, i.e. there exists a great deal of overlap in terms of shared nodes between a pair or set of FFL motifs. *For any node u, we define its motif cluster as the subgraph consisting of the set of nodes and edges participating in a FFL motif with u*.

We find the top 10 motif central nodes and estimate the average participation of high NMC nodes from tier 2 in each others' motif clusters. Figure 4.11(c) shows that, on a scale of 0 to 9, the average participation of high NMC nodes is around 5 which is considered very high in [117], showing that high motif central nodes often participate in the same FFL as other high motif central nodes.

4.2.3.7. Large number of high motif central nodes are directly connected. We create subgraphs consisting of high motif central nodes and edges connecting them. Figure 4.11(d) shows that particularly in mouse and human TRNs, many motif central nodes are directly connected to one another.

4.2.4. Functional Properties of Motif Central Nodes. For each of the four TRN topologies, we first rank the top 10 FFL motif central nodes in tier 2 that do not feature in the top 10 high degree nodes. We then analyze their functional properties in light of their role A and B properties. We do not include role C because they are indicative of regulated rather than regulating entities. We take into account another metric, the motif clustering diversity (MCD) (discussed in Materials and Methods Section 4.1).

4.2.4.1. Motif Clustering Diversity (MCD). Gorochowski et al. argued that motif central nodes within a FFL motif cluster may have a high connectivity but their interactions are often restricted to within the motif cluster, making them unlikely to play a broader role in coordination of many functions across the system. Thus, nodes spanning motif clusters, quantified by a metric called motif clustering diversity (MCD) (defined in Appendix 1 Section 4) of many different types might play a key role in coordination and thus, are important to the system. High MCD value signifies their role as global regulators [117].

4.2.4.2. K-shell decomposition. Kitsak et al. [116] showed that nodes with high $k-$ shell value (defined in Appendix 1 Section 4) makes it a likely candidate for fast spreader of information.

We analyze the functional properties of such high motif central (and low degree central) nodes for human (Table 4.2) TRNs; similar results were also obtained for *E. coli*, *S.cerevisiae* and mouse TRNs and are included in Appendix 1 Section 5. For each high NMC node, we report role A and B centralities, MCD values (higher MCD values correlate with global regulators which in turn correlates with being good information spreaders), and the number of signalling pathways (defined in Appendix 1 Section 4) they participate in. We also report the signalling pathway participation of each node from the KEGG database [118] which is an indirect way of highlighting their information spreading potential. While role A motif centrality closely correlates with the information spreading potential of a node, fault tolerance is better quantified by the role B centralities. Finally, we discuss some findings from published literature on the fault tolerance achieved by some of these high role B motif central nodes.

Human TRN: Functional properties of high motif central nodes in human TRN similarly demonstrate both role A and role B properties (Table 4.2). All of these high motif central (and low degree central) nodes exhibit high MCD values and hence act as global regulators. Since signalling pathways in human TRN are better documented in KEGG, we found a better correspondence of large signalling pathway involvement for these nodes barring some cases such as GATA1. In the following, we similarly report the involvement of seven of these nodes in fault tolerance from published literature as a means for biological validation. Again, we also document the $k-$shell values of the TFs/genes. We observe majority of the nodes reported in Table 4.2 possess the highest $k-$shell value in the network (equal to 11), while GATA1 and GATA3 have $k-$shell values equal to 9 and 10, respectively.

Table 4.2. Functional properties of high motif central nodes in human TRN.

	Roles				
TF/Gene	**A**	**B**	**MCD**	**KEGG Pathways**	**k-value**
ESR1	248	245	12	8	11
HIF1A	130	208	12	11	11
FOS	98	161	12	42	11
HDAC1	182	103	12	14	11
BRCA1	86	144	12	7	11
EGR1	105	163	12	7	11
STAT1	86	108	12	26	11
GATA1	82	94	11	0	9
RB1	88	73	12	89	11
GATA3	67	83	12	5	10

- Transcriptional mediators of cell stress pathways, including $HIF1\alpha$, $ATF4$, and $p53$, are key to normal development and play critical roles in disease, including ischemia and cancer [119] thereby confirming the role B property of $HIF1\alpha$.

- $HDAC1$ links early life stress to schizophrenia-like phenotypes thereby exhibiting role B properties. Early life stress (ELS) is an important risk factor for schizophrenia and authors in [120] show that ELS in mice increases histone-deacetylase ($HDAC$) 1 levels in brain and blood; although altered $Hdac1$ expression in response to ELS is widespread, increased $Hdac1$ levels in the prefrontal cortex are responsible for the development of schizophrenia-like phenotypes. In turn, administration of an $HDAC$ inhibitor ameliorates ELS-induced schizophrenia-like phenotypes.

- BRCA1 regulates oxidative stress and this may be another mechanism in preventing carcinogenesis in normal cells [121] thereby exhibiting role B properties.

- Some of the earliest studies implicating $STATs$ in mediating cell stress responses were performed in cells exposed to UV light. Further analysis showed that STAT1 could be phosphorylated directly by p38 MAPK in vitro. Thus, the MAPK and

STAT pathways appear to converge during periods of cellular stress. In another study, UV light caused STAT1 tyrosine phosphorylation, nuclear accumulation, and DNA binding in keratinocytes. Together, these studies raise the possibility that $STATs$ can be activated in a ligand-independent manner during cellular stress, resulting in the activation of STAT-dependent target genes [122] thereby showcasing their fault tolerance properties.

- STAT1 is a master regulator of Pancreatic Îš-Cell Apoptosis and Islet Inflammation [123] which also demonstrates its fault tolerance property.
- GATA-1 is a master regulator of erythropoiesis. Its role in regulating erythroid-specific genes has been extensively studied, whereas its role in controlling genes that regulate cell proliferation is less understood [124]. Thus, GATA-1 may serve the dual roles of both A and B motif centralities.
- The role of GATA3 protein in the control of the cellular and molecular response of human keratinocytes exposed to a 1 cGy dose of X-rays was investigated in [125] and underlines its role B properties.

4.3. INFERENCES

In this section, we corroborate biological studies that have shown motifs to be robust signal propagation pathways. Our experiments show that dominant 3-node FFL motifs not only contribute to the clustering by forming dense clusters, but also render robustness by creating independent paths. Graph theoretic and simulation experiments on TRN and corresponding bio-WSN topologies show that the motif central nodes participate in bulk of the information flow, even under failure of random and targeted nodes and links. Finally, WSN simulations depict that multiplicity of alternate communication paths in motif structures preserve average shortest path length during node and link failures than that of Erdös Rényi random graphs of the same sparseness. We believe that node motif centrality

(NMC) could be applied in the context of systems biology to gather in-depth knowledge of signal flow dynamics in TRNs. In the future, we shall explore the significance of motif centrality measures in other complex network topologies.

5. TOPOLOGICAL ENHANCEMENT OF TRN BY EDGE REWIRING

We know that TRNs are characterized by the abundance of well-connected nodes, called *hubs*. (It must be mentioned here that certain networks exhibit a property called rich-core by virtue of which there exists a group of densely connected nodes. We discuss in Appendix 2 Section 1 that despite the presence of hubs, TRNs do not possess rich cores.) Our discussion in Section 3.2.7 reveals that TRN graphs are vulnerable to the failure of well-connected nodes. As shown in Table 5.1, over 35% nodes in tiers 2 and 3 have in-degree 1. Such nodes stand a chance of being knocked off the network in the event of failure of well-connected nodes. *In an attempt to overcome this limitation, while preserving its graph properties such as low graph density, motif abundance and scale free out-degree distribution, we perform edge rewiring on existing E. coli and Yeast graphs.*

Table 5.1. Percentage of tier 2 and 3 nodes with in-degree 1 in *E. coli* and Yeast TRN.

	Tier 2		Tier 3	
	No.	%	No.	%
E. coli	47	24.6%	538	36.2%
Yeast	31	35.0%	1554	36.2%

Definition: Edge rewiring is the process of addition and removal of edges in a graph, such that the total number of edges in rewired graph is same as that in original network. In other words, for any input directed graph $G_O(V_O, E_O)$, the rewired graph $G_M(V_M, E_M)$ meets the following three criteria:

1. The number of nodes in original and rewired graphs are equal i.e. $|V_O| = |V_M|$.

2. The number of edges in original and rewired graphs are equal i.e. $|E_O| = |E_M|$.

3. The set difference between the edge set of original and rewired graph is non-empty i.e., $|E_O - E_M| > 0$. This condition ensures that the rewired graph G_M is not completely identical to the original TRN G_O.

Edge rewiring involves two steps: (A) *Edge addition* and (B) *Edge deletion*. Broadly speaking, given an input TRN graph (G_O), α number of edges are added to ensure that all tier 2 and 3 nodes are connected to at least 2 nodes from tiers 1 and/or 2. Following this, α edges must be removed to meet Criterion 2 of edge rewiring. We now discuss the details of the edge addition and edge deletion. All notations used in course of edge rewiring have been enlisted in Table 5.2.

Table 5.2. Table of notations.

Symbol	**Meaning**
$G_O(V_O, E_O)$	Original TRN
$G_A(V_A, E_A)$	Augmented TRN
$G_M(V_M, E_M)$	Rewired TRN
α	No. of edges added during edge addition
$CC(G, u)$	Clustering coefficient of node u in undirected graph H
$\delta_O(u)$	Out-degree of a node u
$\delta_I(u)$	In-degree of a node u in G
Δ_I^j	In-degree distribution of graph G^j
f_i^j	Frequency of nodes with in-degree i in graph G_j (superscript j is optional)
md_I^G	Maximum in-degree of G
ψ^G	Number of FFL motifs in G
$\sigma(e(u, v))$	Edge motif centrality of edge $e(u, v)$
ς^G	List of edge motif centralities in G
t_i^G	List of nodes in i^{th} tier of G
χ^G	List of sink nodes : $t_1^G \cup t_2^G$
E^G	Path count index of G
η^G	Network efficiency of G

5.1. EDGE ADDITION

Edge addition ensures that every tier 2 and 3 node is connected to at least 2 nodes from tier 1 and/or tier 2. We base the edge addition mechanism on *preferential attachment*, which is a growth model for scale free networks like TRN [126]. We introduce the preferential attachment model hereafter.

We have already introduced the notion of preferential attachment growth model in Section 3.2.6. We find the probability of a newly added node to share an edge with existing node u with out-degree $\delta_O(u)$ as:

$$\pi(u) = \frac{\delta_O(u)}{\sum_{w \in V} \delta_O(w)} \tag{5.1}$$

Evidently the use of the preferential attachment growth model will ensure that the resultant graph after edge addition, termed *augmented TRN* (G_A), will continue to retain the scale-free out-degree distribution (one of the key TRN properties we intend to preserve).

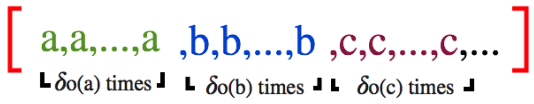


Figure 5.1. Preferential attachment list L_U.

Algorithm description: Edge addition algorithm generates a *preferential attachment list* L_U consisting of tier 1 and 2 nodes. The frequency of each node u in L_U is equal to its out-degree $\delta_O(u)$ (Lines 2 - 8), as also illustrated in Figure 5.1. For each tier 2 and 3 node v in list L_V (consisting of nodes with in-degree $\delta_I < 2$), we introduce an edge between v and a randomly selected node u from L_U, based on the formulation of preferential attachment (Eq. 5.1), as shown in Lines 9 - 15. Hence, all the nodes in tiers 2 and 3 of augmented TRN G_A have in-degree of at least 2.

Algorithm 1 Edge Addition Algorithm

```
 1: procedure
 2:     $\alpha = 0, L_U = \emptyset, L_V = \emptyset$
 3:     $G_A = G_O$
 4:     for $u \in V_A$ do
 5:         if $\delta_I(u) < 2$ and $u \notin$ tier 1 of $G_A$ then
 6:             $L_V.append(u)$
 7:         if $\delta_O(u) > 0$ then
 8:             $L_U.append([u$ for $i = 1$ to $\delta_O(u)])$
 9:     for $v \in L_V$ do
10:         while True do
11:             $u = random(L_U)$
12:             if $e(u, v) \notin E_A$ then
13:                 $E_A = E_A \cup e(u, v)$; $L_U.append(u)$
14:                 $\alpha = \alpha + 1$
15:                 break
```

Time Complexity: The for loop in Line 4 iterates $|V_A|$ times. The random function used in Line 11 selects an unique element each time from L_U. Since the minimum δ_I of any node v in L_V is 1, the while loop (Line 10) iterates at most twice so we encounter u such that $e(u, v) \notin E_A$. Thus, time complexity of the edge addition algorithm is $O(2 \times |V_A|) = O(|V_A|)$.

5.2. EDGE DELETION

In order to meet criterion 2 of edge rewiring, we need to remove $\alpha = |E_A| - |E_O|$ edges from G_A. It is imperative to maintain *three constraints* during edge deletion.

5.2.1. 2-connectivity. Edges can only be removed from nodes with in-degree $\delta_I > 2$, since the removal of any other edge would cause nodes from tiers 2 and 3 in rewired TRN G_M to have in-degree less than 2. As an example, Figure 5.2 shows the in-degree distribution of Yeast TRN subgraph of 400 nodes, where the gray area represents edges participating in nodes with $\delta_I \leq 2$. Edges can only be removed from the white region under the curve. Evidently, this needs to be an important consideration while carrying out edge deletion from the augmented TRN.

5.2.2. Dynamics of Degree Distribution. Deletion of an edge with in-degree $i + 1$ decreases the frequency of in-degree $i + 1$ (denoted by f_{i+1}) by 1 and increases the frequency of in-degree i (denoted by f_i) by 1 in the in-degree distribution curve (Figure 5.3). Therefore, the number of edges of any in-degree i available for removal depends on the frequency of available edges of in-degree i as well as $i + 1$.

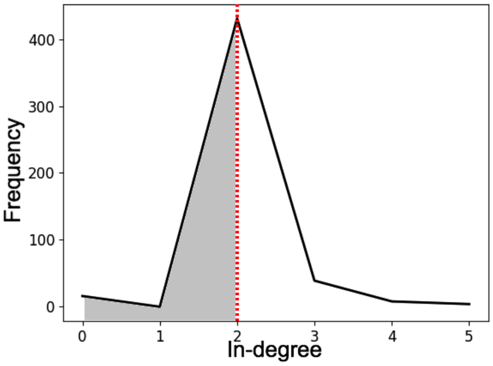


Figure 5.2. In-degree distribution of augmented Yeast TRN subgraph of 400 nodes.

5.2.3. Robustness Due to FFL Motifs. FFL motifs have been proven to play a significant role in rendering topological robustness to TRN in the following two ways [127]:

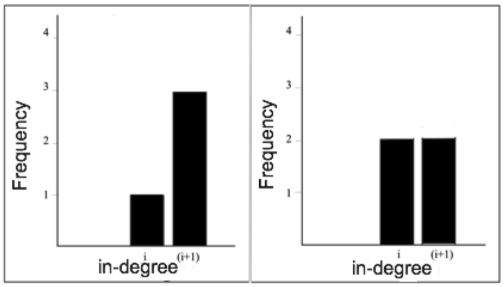


Figure 5.3. Dynamics of in-degree distribution due to edge deletion.

5.2.3.1. Robustness due to independent paths. Two paths between a node pair are called independent if they contain no common nodes, except source and destination nodes. For instance in graph G, shown in Figure 5.4(a), there are 3 independent paths between nodes 1 and 2, namely $\{1 \rightarrow 3 \rightarrow 2\}$, $\{1 \rightarrow 2\}$ and $\{1 \rightarrow 4 \rightarrow 2\}$. According to *Menger's theorem on vertex connectivity*, the minimum number of vertices whose removal disconnects two nodes is equal to the maximum number of pairwise vertex-independent paths between them [112]. For example in graph G, at least two nodes (3 and 4) must be removed to disconnect nodes 1 and 2. (Note that $\{1 \rightarrow 2\}$ is also an independent path but, being an edge, it has no intermediate nodes to disconnect.) Since the FFL motif contains two independent paths connecting source node S to target node T (as shown in Figure 5.4(b)), *abundance of FFLs ensures topological robustness in TRNs by offering multiple alternative communication pathways*.

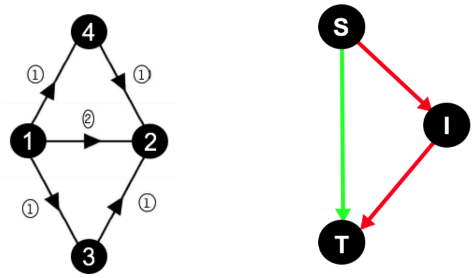


Figure 5.4. Robustness due to FFL motifs. (a) Example graph G with edge motif centralities shown in circles (b) Two paths marked in different colors from source S to target T in FFL.

5.2.3.2. Increase in shortest path length. Node or link failures in any graph may increase the shortest path length between pairs of existing nodes, or it may make them unreachable from one another. In case of FFLs, the failure of direct links between source S and target T causes the shortest path length between S and T to increase only by a single hop. *Hence, abundance of FFLs makes TRNs robust by minimizing the increase in shortest path length during failures of nodes*.

Combining points (a) and (b) we infer that FFL motifs contribute to topological robustness of TRNs by providing multiple, short communication pathway alternatives, despite node failures.

In order to meet the core requirements of edge rewiring, it is crucial that we incorporate the above constraints in the edge deletion algorithm. For instance, as per the *2-connectivity* constraint, each tier 2 and 3 node has in-degree 2. Consequently, *2-connectivity* would make the rewired TRN more robust against failure of well-connected nodes. Similarly, *dynamics of degree distribution* should help delete edges from G_A, while preserving the in-degree distribution of TRN. Finally, the notion of *robustness due to FFL motifs* makes it imperative to preserve them in course of edge deletion.

In the following sections, we first discuss a simply greedy edge deletion approach and highlights its drawbacks. We then propose an improved dynamic edge deletion approach.

5.3. GREEDY EDGE DELETION

In this algorithm, we remove α number of edges with minimum motif centrality from G_A. Going back to the example of graph G shown in Figure 5.4, we prefer to preserve $e(1, 2)$ because $\sigma(e(1, 2))$ is the highest. We also attempt to incorporate 2-connectivity and dynamics of degree distribution.

Algorithm 2 Greedy Edge Deletion

```
procedure
    for $i = 3$ to $md$ do
        Calculate $T_i$ using Eq. 5.2 , $j = 0$
        Sort edges in $dlist_i$ in non-decreasing order of motif centrality $\varsigma$
        $dlist = sort(dlist, \varsigma, i)$
        $R = \emptyset$
        while $T_i > 0$ and $j < |dlist_i|$ do
            $dlist_{i,j}$ is the $j^{th}$ edge of $dlist_i$
            $e(u, v) = dlist_{i,j}$
            if $e \in E_A$ and $\delta_I(v) > 2$ then
                $E_A = E_A - e$;
                $T_i = T_i - 1$;
                $R = R \cup e$
            $j = j + 1$
        $dlist = updateDlist(dlist, R)$
```

Algorithm description: Greedy edge deletion takes as input (i) the augmented TRN $G_A(V_A, E_A)$, (ii) maximum in-degree of G_A, (denoted by md), (iii) a data structure $dlist$, where each entry $dlist_i$ contains all edges incident to nodes with in-degree i arranged in the non-decreasing order of σ-value (as shown in Figure 5.5) and (iv) list of motif centrality of all edges in G_A, denoted by $\varsigma = (\sigma(e_1), \sigma(e_2), \cdots, \sigma(e_{|E_A|}))$. The for loop in Line 2 traverses 3 to md. (Note that removing any edge from node with $\delta_I \leq 2$ will violate 2-connectivity constraint.) For each in-degree i, we remove T_i edges from $dlist_i$. The process of determination of T_i (Line 3), for all $i < md$, has been discussed hereafter. The while loop stops when T_i edges of $dlist_i$ are removed or all edges in $dlist_i$ have been traversed. Each removed edge e is added to a removed list R (Lines 7 - 14). In function *updateDlist* we add each $e(u, v)$ in removed list R to $dlist_{\delta_I(v)}$ (Line 15). Finally, the edges in each $dlist_i$ are re-sorted in the non-decreasing order of σ for subsequent i values (Line 5).

Determination of T_i: An important step in the edge deletion is the determination of T_i, which denotes the number of edges of in-degree i to be removed. Note that i in T_i ranges from 3 to md because we are only removing edges from nodes with in-degree greater than 2. The value of T_i is dictated by frequency of available edges of in-degree i and $i + 1$ (denoted by f_i and f_{i+1}, respectively). This is due to the fact that the removal of an edge with in-degree $i + 1$ causes an increase in the frequency of nodes with in-degree i (as discussed in Section 5.2.2). Therefore, T_i is given by:

$$T_i = \frac{a \times f_i + (1 - a) \times f_{i+1}}{D} \times \alpha, \tag{5.2}$$

where (i) a is a scale constant lying between 0 and 1 which determines the weightage of f_i and f_{i+1} and (ii) D is the normalizing constant given by $D = \Big(a \sum_{i=3}^{md} f_i + (1 - a) \sum_{i=4}^{md} f_{i+1}$ $\Big)$. Note that $f_{i+1} = 0$ when $i = md$. In our experiments, we consider $a = 0.5$.

Time complexity: The $i-$loop in Line 2 also iterates $md-2$ times. The determination of T_i step has complexity $O(1)$. In Line 5, sorting the edges in the non-decreasing order of ς incurs complexity $O(|E_A|\ lg\ |E_A|)$. In Lines 7 - 14, the while loop removes T_i edges, incurring cost of $O(|E_A|)$. Finally, the function *updateDlist* in Line 15 has a cost of $O(|E_A|)$. Thus, the total complexity of the greedy edge deletion algorithm is $O((md - 2) \times |E_A|\ lg\ |E_A|) = O(|E_A|\ lg\ |E_A|)$.

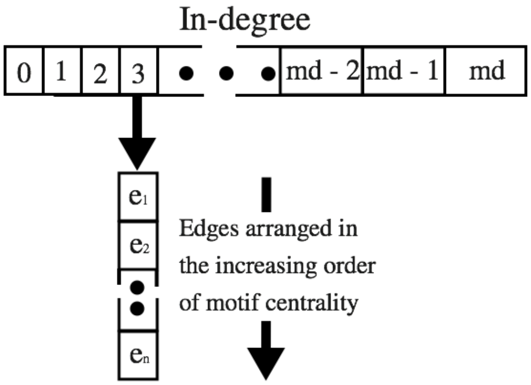


Figure 5.5. Data structure $dlist$ for greedy edge deletion.

5.4. DYNAMIC EDGE DELETION

There exists few important issues with the greedy edge deletion. We introduce a notion called *obtainability*, which is crucial for understanding one of the issues in the greedy edge deletion approach, leading up to dynamic edge deletion.

5.4.1. Obtainability. Let us consider directed graphs G^1 and G^2 such that $V(G^1) = V(G^2)$ and $E(G^2) \subset E(G^1)$. Now we define in-degree distribution of graphs G^1 and G^2 as $\Delta_I^1 = (f_1^1, f_2^1, \cdots, f_{md}^1)$ and $\Delta_I^2 = (f_1^2, f_2^2, \cdots, f_{md}^2)$, respectively. Here md is the maximum in-degrees of G^1 and f_i^j *is the frequency of nodes with in-degree i in graph* G^j*, respectively.*

Definition 4: Given two directed graphs G^1 and G^2 with in-degree distributions Δ_I^1 and Δ_I^2, we define Δ_I^2 as *obtainable* from Δ_I^1 if it is possible to obtain G^2 by deletion of one or more edges from G^1.

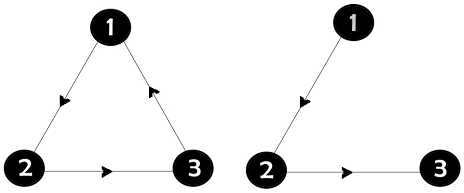


Figure 5.6. Example of obtainability. Graph G^1 (Left) and G^2 (Right).

For instance, in Figure 5.6 we consider two directed graphs G^1 and G^2. G^2 is formed by deletion of edge $e(1,3)$ in G^1. In such as case, $\Delta_I^2 = (1,2)$ is said to be *obtainable* from $\Delta_I^1 = (0,3)$.

Effect of obtainability on in-degree distribution: Let us consider three directed graphs G^1 (Figure 5.7 (left)), G^{in} (Figure 5.7 (middle)) and G^2 (Figure 5.7 (right)), such that G^2 and G^{in} are obtainable from G^1 by deletion of 3 edges. In Figure 5.7, we observe that $\Delta_I^1 = (0,3,1)$, $\Delta_I^{in} = (0,4,0)$ and $\Delta_I^2 = (2,2,0)$. We intuit that exists a relationship between Δ_I^1 and Δ_I^2.

From our discussion in Section 5.2.2, we know that deletion of edge with in-degree $i+1$ decreases the frequency of in-degree $i+1$ by 1 and increases the frequency of in-degree i by 1 in the in-degree distribution curve. Given any in-degree K (where $0 \leq K \leq md$), the removal of edges from nodes with in-degrees $K+1$ to md must affect f_K.

Let us analyze the relationship between f_K^{in} and f_K^2 w.r.t f_K^1 for any arbitrary value of K (**say** $K = 0$).

- In Figure 5.7. if $e(2,4)$ is removed from G^1 we obtain G^{in}. As $e(2,4)$ is removed, $f_2^{in} = f_2^1 - 1 = 1 - 1 = 0$ and $f_1^{in} = f_1^1 + 1 = 3 + 1 = 4$. Any change in f_2 and f_1 flow into f_0, i.e. $f_0^{in} = f_0^1 + \sum_{i=1}^{md}(f_i^1 - f_i^{in}) = 0 + (1-0) + (3-4) = 0$.

- Similarly, if $e(3,1)$ and $e(1,2)$ are removed from G^{in}, we obtain G^2. Then, $f_0^2 = f_0^1 + \sum_{i=1}^{md}(f_i^1 - f_i^2) = 0 + (1-0) + (3-2) = 2$.

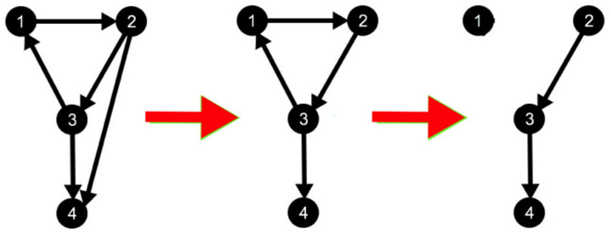


Figure 5.7. Edge deletion and obtainability. Directed graphs G^1 (left), intermediate graph G^{in} (middle) and G^2 (right).

Thus, if Δ^2 is obtainable from Δ^1, the relationship between f_K^1 and f_K^2 is given by $f_K^2 = f_K^1 + \sum_{i=K+1}^{md}(f_i^1 - f_i^2)$, for any $0 \leq K \leq md$.

Note: In Figure 5.7, since G^2 is obtainable from G^{in}, the same relationship exists in the in-degree distribution of G^{in} and G^2 as well. Let us formalize this idea in the following lemma.

Δ_I^2 is *obtainable* from Δ_I^1 if $f_K^2 = f_K^1 + \sum_{i=K+1}^{md}(f_i^1 - f_i^2)$, where $f_i^2 \leq f_i^1$ and $0 \leq K \leq md$.

Let us consider in-degree distribution of G^1, $\Delta_I^1 = f_0^1, f_1^1, \cdots f_{md}^1$, and in-degree distribution of G^2, $\Delta_I^2 = f_0^2, f_1^2, \cdots f_{md}^2$.

Let us begin with graph G^1 and remove $(f_{md}^1 - f_{md}^2)$ edges incident to nodes of in-degree md. The resultant graph has distribution $(f_1^1, f_2^1, \cdots, f_{md-1}^2, f_{md}^2)$, where $f_{md-1}^2 = f_{md-1}^1 + (f_{md}^1 - f_{md}^2)$.

If, in the same graph, we remove $(f_{md-1}^1 - f_{md-1}^2)$ edges incident to nodes of in-degree $md - 1$, we obtain distribution $(f_1^1, f_2^1, \cdots, f_{m-2}^2, f_{m-1}^2, f_{md}^2)$, where $f_{md-2}^2 = f_{md-2}^1 + (f_{md-1}^1 - f_{md-1}^2) + (f_{md}^1 - f_{md}^2) = \sum_{i=md-1}^{md}(f_i^1 - f_i^2)$.

Finally, for all $i = md - 2, md - 1, \cdots, K + 1, K$, if we continue to remove $(f_i^1 - f_i^2)$ edges of in-degree i, we obtain in-degree distribution $(f_0^1, f_1^1, \cdots, f_K^2, f_{K+1}^2 \cdots, f_{md}^2)$, where $0 \leq K \leq md$ and $f_K^2 = f_K^1 + \sum_{i=K+1}^{md}(f_i^1 - f_i^2)$.

5.4.2. Issues with Greedy Edge Deletion. Let us now understand the following drawbacks with the greedy edge deletion algorithm.

5.4.2.1. Determination of optimal T_i. The formulation of T_i in Eq. 5.2 is a greedy approach, where T_i is proportional to the f_i and f_{i+1}. However, from the precept of obtainability (discussed in Section 5.4.1) it is clear that f_i is affected by any change in $f_{i+1}, f_{i+2}, \cdots, f_{md}$. It follows that the determination of T_i during edge deletion should ideally incorporate the notion of obtainability. Therefore, we now formulate determination of T_i as a *nonlinear optimization problem* with the objective of preserving in-degree distribution of original TRN (G_O) by minimizing the squared error between the in-degree distribution of G_O (denoted by Δ_I^O) and G_M (denoted by Δ_I^M), with obtainability as one of the constraints.

$$\underset{\Delta_I^M(3:)}{argmin} \sum_{i=3}^{md} (f_i^M - f_i^O)^2 \tag{5.3}$$

$$s.t. \quad \sum_{i=0}^{md} f_i^M \times i = \sum_{i=0}^{md} f_i^O \times i \tag{5.4}$$

$$f_2^A + \sum_{i=3}^{md} (f_i^A - f_i^M) = f_2^M \tag{5.5}$$

$$f_i^M \leq f_i^A \quad \forall i = 3 \quad to \quad md \tag{5.6}$$

- The objective function (Expression 6.6) minimizes the squared error between the in-degree distribution of original and rewired TRN. Note that we do not change frequency of nodes with in-degrees 0 to 2, as we ensure that all tier 2 and 3 nodes have in-degree at least 2. This optimization returns $\Delta_I^M(3:) = f_3^M, f_4^M, \cdots, f_{md}^M$. We then calculate $T_i = f_i^O + \sum_{j=i+1}^{md}(f_j^O - f_j^M) - f_i^M$ ($\forall i = 3, 4, \cdots, md$).

- Given any directed graph $G(V, E)$, $|E| = \sum_{i=0}^{md} f_i \times i$ Constraint 5.4 ensures that the number of edges in G_M and G_O are be same (Criterion 2 of edge rewiring).

- Constraint 5.5 guarantees obtainability of rewired TRN from augmented TRN G_A. In absence of this constraint, we cannot ensure that Δ_I^M returned by optimization is a valid in-degree distribution obtainable from G_A.

- As a consequence of edge addition, the frequency of nodes with in-degree 2 increases in rewired TRNs (as shown in Figure 5.8). Additionally, the frequency of nodes with in-degree greater than 2 in rewired TRN is less than that in original TRN graph. Constraint 5.6 reflects this redistribution of node frequency among in-degrees in rewired TRNs.

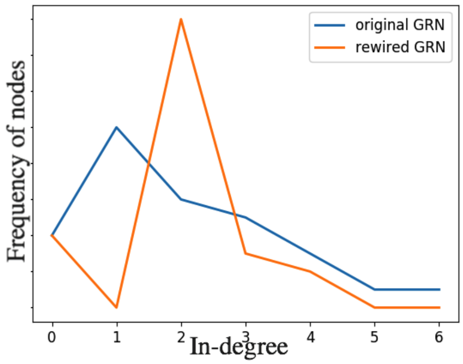


Figure 5.8. Generic form of in-degree distribution of original and rewired TRN.

5.4.2.2. Change in motif centrality. In Figure 5.9, we consider an edge $e(u, v)$ that participates in L number of FFL motifs. We intuit that the removal of $e(u, v)$ reduces FFL motif count of graph G by L. We formalize this notion in Lemma 5.4.2.2.

Given any directed graph $G(V, E)$ with FFL motif count $\psi(G(V, E))$, the number of FFL motifs $\psi(G(V, E))$ lost due to elimination of any $e \in E$ from G exactly equals the edge motif centrality of e given by $\sigma(e(u, v))$, i.e. $\psi(G(V, E)) - \psi(G(V, E - \{e\})) = \sigma(e(u, v))$.

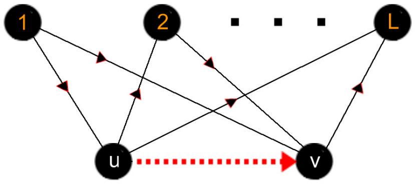


Figure 5.9. Change in motif centrality due to edge deletion. Edge motif centrality $\sigma(G, e(u, v)) = L$; Removal of $e(u, v)$ transforms L FFL motifs into open triplets.

From definition 2, we know that $e(u, v)$ participates in $\sigma(e(u, v))$ FFL triangles. It follows that removal of edge e transforms $\sigma(e(u, v))$ FFLs into open triplets (Figure 5.9 shows the removed edge $e(u, v)$ in red), causing the loss of $\sigma(e(u, v))$ FFL motifs from G. Therefore, $\psi(G(V, E)) - \psi(G(V, E - \{e\})) = \sigma(e(u, v))$.

Given any directed graph G, removal of the edge with the lowest σ during each step of edge deletion preserves the optimal number of FFL motifs.

Let us consider a directed graph G. Let the graph with maximum FFL motif count obtained after deletion of i edges from G be denoted by ${}_iG^l$. At this point, let us now consider two distinct alternatives of edge removal on ${}_iG^l$:

1. Delete the edge with lowest σ (denoted by ${}_ie^l$) from ${}_iG^l$. From Lemma 5.4.2.2, we know that $\sigma({}_ie^l)$ FFL motifs will be lost. Therefore, we have $\psi({}_{i+1}G^l) = \psi({}_iG^l) - \sigma({}_ie^l)$.

2. Delete any edge besides the one with lowest σ (denoted by ${}_i\widehat{e^l}$) from ${}_iG^l$. We have $\psi({}_{i+1}\widehat{G^l}) = \psi({}_iG^l) - \sigma({}_i\widehat{e^l})$.

Since σ is the lowest, $\sigma({}_i\widehat{e^l}) < \sigma({}_ie^l)$, $\psi({}_{i+1}G^l) > \psi({}_{i+1}\widehat{G^l})$. Therefore, the removal of α edges with least σ at each step preserves the optimal FFL motifs.

From Lemma 5.4.2.2 and Corollary 5.4.2.2, we deduce that (i) deletion of each edge e reduces the FFL motif count of the graph by $\sigma(e)$ and (ii) optimal FFL motifs are preserved by deleting the edge with the lowest σ. Now recall that the greedy edge deletion approach does not update the σ of edges after each deletion. Since it does not necessarily delete the edge with the lowest σ, it may not preserve the optimal number of FFL motifs, necessitating an improved dynamic edge deletion approach.

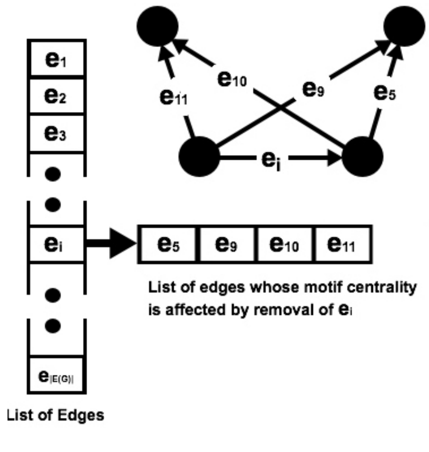

e1
e2
e3
ei
e|E(G)|
List of Edges
e11
e10
e9
e5
ei
e5
e9
e10
e11
List of edges whose motif centrality
is affected by removal of ei

Figure 5.10. Dynamic Edge Motif List DL.

5.4.3. Dynamic Edge Deletion Algorithm. In this algorithm, the key objectives are to (i) delete optimal $\alpha = \sum_{i=3}^{md} T_i$ edges (where T_i is determined using non-linear optimization), and (ii) preserve maximum number of FFL motifs at each edge deletion step (using Lemma 3). We solve the nonlinear least squares optimization using the Python SciPy library [128]. We discuss the steps in dynamic edge deletion algorithm.

Algorithm description: Dynamic edge deletion algorithm maintains a list of motif centrality of edges ς and a Dynamic Edge Motif List DL (Figure 5.10), where, for each edge e, there is a list of edges that share a FFL motif with e. Given the value of T (determined by nonlinear least squares optimization) and α, the algorithm finds the edge with the lowest σ by invoking $findLowestMotifEdge(\varsigma)$, $e^l(u, v)$ (Line 4). The edge $e^l(u, v)$ is eligible for deletion if it meets two conditions: (1) $\delta_I(v)$ exceeds 2 and (2) $T_{\delta_I(v)}$ is more than 0. If the eligibility conditions are satisfied, $T_{\delta_I(v)}$ is decremented and e^l is removed from G_A (Lines 5 - 7). Finally, function $adjustEMC$ is invoked to update the σ of all edges affected by the deletion of e^l. Specifically, σ of each edge in DL_{e^l} is decremented by 1. This update ensures that we remove α least motif central edges e^l in each step to preserve the maximum FFL motifs in G_M (Line 8). The algorithm terminates when α edges are removed.

Algorithm 3 Dynamic Edge Deletion

```
1: procedure
2:     $i = 0$
3:     while $i < \alpha$ do
4:         $e^l(u, v) = findLowestMotifEdge(\varsigma)$
5:         if $T_{\delta_I(v)} > 0$ and $\delta_I(v) > 2$ then
6:             $T_{\delta_I(v)} = T_{\delta_I(v)} - 1$
7:             $G_A.remove\_edge(e^l)$
8:             $\varsigma = adjustEMC(DL, e^l, \varsigma)$
9:             $i = i + 1$
```

Time complexity: The while loop iterates α times. Within each iteration, function $findLowestMotifEdge$ traverses the edge-list finds the edge with lowest σ in $O(|E_A|)$ time, and $adjustEMC$ decrements σ of each edge in DL_{e^l} by 1 in $O(|E_A|)$. Therefore, the overall complexity of dynamic edge deletion is $O(\alpha \times |E_A|)$.

5.5. COMPARATIVE ANALYSIS OF G_O AND G_M

We first compare the original and rewired TRNs across all graph orders in terms of the four graph metrics. Since the motivation behind edge rewiring is to preserve the graph attributes of original TRN, while remedying its vulnerability to failure of well-connected nodes, we explore how similar original and rewired TRNs are to one another. The results in Table 5.5 present the combined average scores the four graph metrics of both TRNs.

Graph generation: For both *E. coli* and Yeast, we utilize GeneNetWeaver to generate 50 subgraphs each of sizes 100, 200, 300, 400 and 500 nodes. Each generated subgraph has approximately the same graph density as original *E. coli* and Yeast TRN topology. For each original *E. coli* and Yeast TRN subgraph we use Python Networkx library [129] to generate an E-R random graphs of approximately same graph density.

5.5.1. Diameter. It is the measure of the longest shortest path between any pair of nodes in a given graph. Since, TRNs are weakly connected directed graphs, we calculate the diameter of undirected TRN subgraphs. Results show that the average diameter of undirected rewired TRNs are slightly lower than original TRN subgraphs.

5.5.2. Average Clustering Coefficient (ACC). ACC of dynamically rewired TRNs is the highest, followed by greedily rewired TRNs. From our discussion in Section 3.2.4, we know that high ACC warrants high FFL motif preservation.

5.5.3. Assortativity. It is a measure of the tendency of nodes to attach to other similar nodes. The assortativity coefficient [130] of any directed graph G is based on the Pearson Correlation Coefficient. It is calculated as:

$$r = \frac{\sum_{i,j}(A_{i,j} - \frac{k_i k_j}{2m}) \times x_i x_j}{\sum_{i,j}(\kappa_{i,j} k_i - \frac{k_i k_j}{2m}) \times x_i x_j} \tag{5.7}$$

Here A is the graph adjacency matrix of directed graph G, k_i is degrees of node i, $\kappa_{i,j}$ is Kronecker function, x_i is a scalar associated with node i and m is the total number of edges. The value of r is a score ranging from -1 (disassortative) to +1 (assortative). Scale-free networks are usually disassortative because the preferential attachment growth model causes edge addition between a poorly connected and well-connected node [130]. Although the edge addition algorithm follows preferential attachment growth model, edge deletion causes some well-connected nodes to lose poorly-connected neighbors. Thus, the rewired TRNs are slightly more assortative than the original TRN subgraphs.

5.5.4. Degree. Average degree of dynamically rewired TRN is slightly more than original and greedily rewired TRNs.

Note: The average diameter and degree tend to increase with graph order, we normalize each average diameter and degree score of each subgraph by the order of graph.

From the summary of scores of four graph metrics in Table 5.3, we conclude that original and rewired TRNs are topologically similar.

Table 5.3. Graph properties of original and rewired TRN. Average (a) diameter (b) clustering coefficient (c) assortativity (r) and (d) degree

	Diam.	**ACC**	**r**	**Deg.**
Original	0.022	0.140	−0.286	0.034
Greedy	0.020	0.187	−0.233	0.034
Dynamic	0.020	0.194	−0.231	0.036

5.6. EXPERIMENTAL RESULTS

We analyze the topology of original and rewired *E. coli* and Yeast TRN, both greedy and dynamic, in light of certain graph and simulation experiments. We compare the performance of TRN against E-R random graphs of roughly same graph density as TRN.

Reason for using E-R random graphs as benchmark: For TRN-inspired WSNs our initial objective was to compare the proposed topologies to other standard topologies of similar graph density. Unlike other standard topologies like k-connected or scale free, it is possible to control the density of the random topologies by regulating the probability of edge existence *p*.

5.6.1. Graph Experiments. Let us analyze the results for the graph experiments.

5.6.1.1. Degree distribution. We show the average in and out-degree distributions of original and rewired *E. coli* and Yeast TRN subgraphs. For each subgraph of original, greedy and dynamically rewired TRN, in and out degree distributions are normalized by degree sum. Figure 5.11(a) and 5.11(b) show the mean curves of normalized in and out degree distribution of all TRN subgraphs.

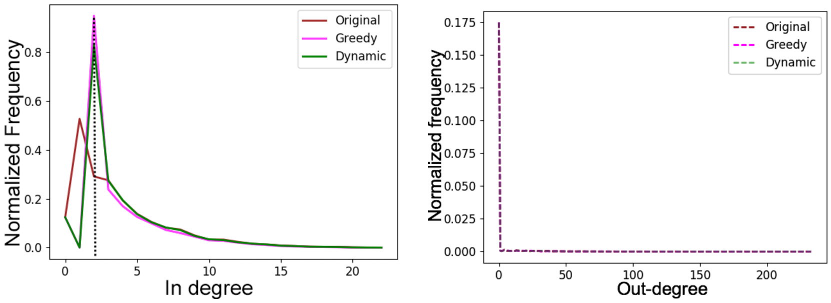


Figure 5.11. Degree distribution due to rewiring. (a) Normalized average in-degree and (b) Normalized average out-degree distribution of *E. coli* and Yeast TRN subgraphs. The dotted line shows that in-degree distribution of rewired TRN peaks at in-degree 2.

In Figure 5.11(a) we observe that the in-degree distribution curves of greedily and dynamically rewired TRNs peak at in-degree 2 (shown in dotted line). This is a direct consequence of edge rewiring where all tier 2 and 3 nodes are connected to at least 2 nodes from tiers 1 and 2. Recall, the purpose of minimization of squared error discussed in Section 5.4.2.1 is to ensure that the in-degree distribution of dynamically rewired TRN is close to that of original TRN. We measure the degree of deviation of in-degree distribution curve of any rewired subgraph from that of original TRN subgraph using the notion of Root Mean

Square Deviation (RMSD):

$$RMSD(\Delta_I^1, \Delta_I^2) = \sqrt{\frac{\sum_{k=3}^{md}(f_k^1 - f_k^2)^2}{md - 2}} \tag{5.8}$$

We observe that average $RMSD(\Delta_I^{G_O}, \Delta_I^{G_M})$ for greedily and dynamically rewired TRN are 0.01128 and 0.00126, respectively. Thus, deviation in in-degree distribution of dynamically rewired is nearly 10 times less than that of original TRN subgraphs, bearing out the importance of the nonlinear optimization employed during dynamic edge deletion.

Figure 5.11(b) shows that dynamically and greedily rewired TRN subgraphs both preserve the out-degree distribution of original TRN subgraphs.

5.6.1.2. Motif preservation. We compare the number of FFL motifs preserved by all the topologies. The results (Figure 5.12(a) and 5.12(b)) show that dynamically rewired *E. coli* and Yeast TRN, patently preserve the highest number of FFL motifs, followed by greedily rewired TRNs.

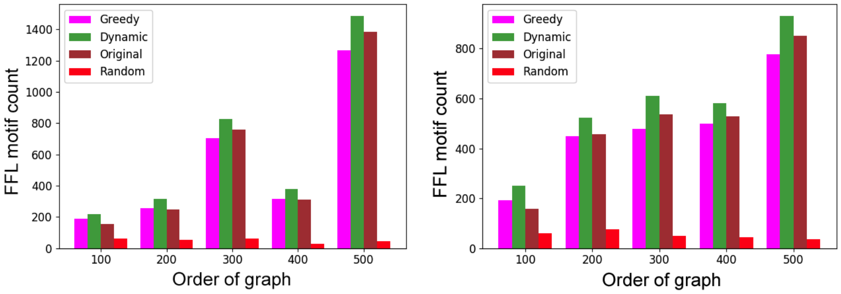


Figure 5.12. FFL motif preservation due to rewiring. Original and rewired (a) *E. coli* and (b) Yeast TRNs and E-R random graphs.

5.6.1.3. Robustness. We compare the robustness of the original TRN, rewired TRN and E-R random graphs in terms of following metrics: average *network efficiency*, *path count*, *number of connected components* and *size of largest connected component*. Studies show that E-R random graphs exhibit robustness against targeted failures, whereas original TRNs, like all scale free graphs, are robust against random failures. We expect the rewired TRNs to retain the best traits of E-R random graphs and TRNs [83].

Normalization: The scores for each of the four robustness metrics increase with the order of input graph. Therefore, in order to obtain a unified metric for all graph orders, the score for any graph is normalized by the order of the graph.

Failure of Random and Targeted nodes: We remove 4%, 8%, 12%, 16% and 20% nodes. Given any input graph, we consider two kinds of failures:

- *Random Failure*: Nodes to be removed are randomly chosen from the node set of the graph.
- *Targeted Failure*: We term nodes in tiers 1 and 2 with out-degree greater than median out-degree of the graph, as **hubs**. Nodes to be removed are chosen from the hub set.

For the following experiments, the edge directions in each subgraph are reversed; *tier 2 and 3 nodes are source nodes and hubs are the destination nodes.* The results reflect the combined average of both *E. coli* and Yeast TRN subgraphs.

5.6.1.4. Network efficiency. *It is a measure of the average shortest path length between all pairs of source and sink nodes.* For any directed graph $G(V, E)$, it is given by:

$$\mathcal{E} = \frac{1}{\phi_E} \sum_{\substack{u \in V, \\ v \in \chi, u \neq v}} \frac{1}{d(u, v)} \tag{5.9}$$

Here (i) χ is the set of sinks, (ii) ϕ_E is the number of source-sink pairs calculated as $|(u, v) : u \in V, v \in \chi, u \neq v|$, and (iii) $d(u, v)$ is the shortest path length between any node pair $u, v \in V$.

Recall that we discuss in Section 5.2.3 that the abundance of FFL motifs provide multiple short path alternatives in TRNs. While comparing the network efficiency $\mathcal{E}$ of the four topologies, we consider source-to-sink shortest paths of a limited number of hops. As an example, we show $\mathcal{E}$ w.r.t paths of length at most 4 in Figure 5.13(a) and 5.13(b). Results show that dynamically rewired TRN subgraphs exhibit significantly better $\mathcal{E}$ than original and greedily rewired TRNs.

Note that random graphs exhibit better $\mathcal{E}$ than TRN graphs as the percentage of targeted nodes failure tends to 20%. This is due to the fact that the number of source-destination paths in scale free networks like TRNs decrease, as the hub nodes are knocked off the graph. In contrast, majority of nodes in E-R random graphs have average degree. Thus E-R random graphs are less affected by targeted node failures, making them exhibit steady $\mathcal{E}$ during targeted failures.

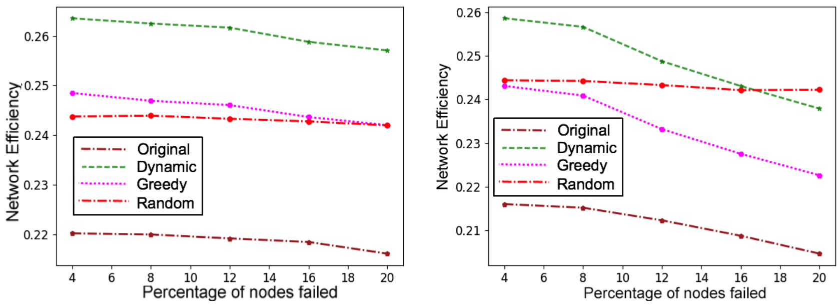


Figure 5.13. Average network efficiency. (a) Random and (b) Targeted node failure.

5.6.1.5. Path count index η. *It measures the average number of simple paths between all pairs of source and sink nodes that are connected by at least one path.* For any directed graph $G(V, E)$, it is given by:

$$\eta = \frac{1}{\phi_\eta} \sum_{\substack{u \in V, \\ v \in \chi, u \neq v}} P(u, v) \tag{5.10}$$

In Eq. 5.10, (i) $P(u, v)$ is the number of simple paths between any node $u \in V$ and $v \in V$ and (ii) ϕ_η is the number of source-sink pairs connected by at least one path i.e. $|(u, v) : u \in V, v \in \chi, u \neq v, G.has\_path(u, v)|$.

Figures 5.14(a) and 5.14(b) show that η value of dynamically rewired TRN subgraphs is the highest for both targeted and random node failures. Once again E-R random graphs, by virtue of its average degree nodes, exhibit steady η during targeted failures.

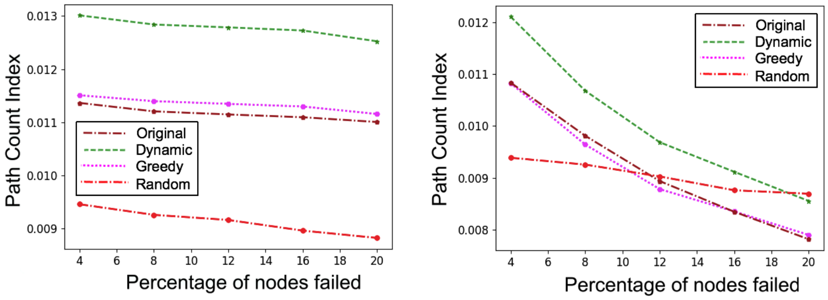


Figure 5.14. Average normalized path count. (a) Random and (b) Targeted node failure.

5.6.1.6. Number of connected components. Node failures will cause any graph to disintegrate into multiple connected components. Since the basis of robustness in our work is the ability of a graph to stay connected despite failures, we compare the number of connected components in the TRN and E-R random subgraphs during random and targeted node failures.

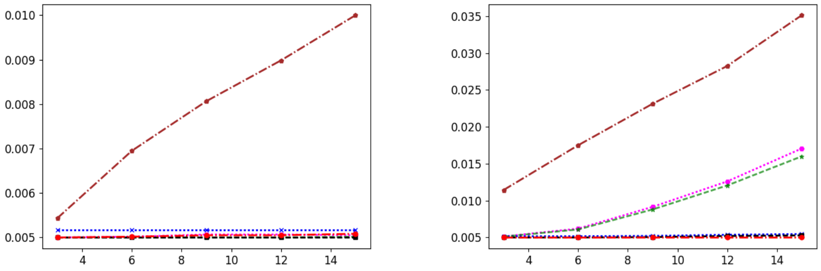


Figure 5.15. Normalized number of connected components. (a) Random and (b) Targeted node failure.

Figure 5.15(a) shows that rewired TRNs disintegrate into the fewest component during random failures. Once again, the E-R random graphs exhibit least vulnerability to targeted node failure (Figure 5.15(b)).

5.6.1.7. Size of largest connected component. We explore how the size of largest connected component is affected by random and targeted node failures. We observe that the size of largest connected component of rewired TRN is marginally better than E-R random graphs and original TRNs, in the event of random node failure (Figure 5.16(a)).

The TRN graphs, due to its susceptibility to failure of targeted nodes perform poorly compared to E-R random graphs, albeit the rewired TRNs still retain larger giant components compared to original TRN subgraphs (Figure 5.16(b)).

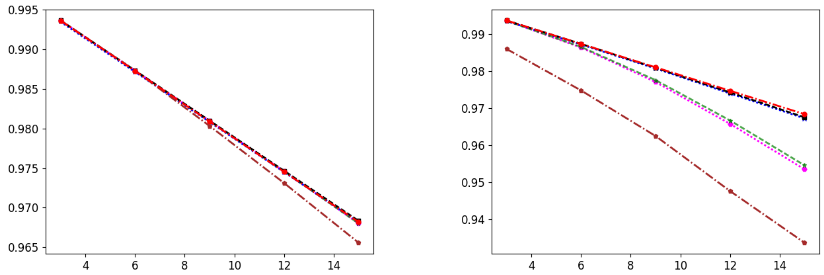


Figure 5.16. Size of largest connected component. (a) Random and (b) Targeted node failure.

From the above experiments it is clear that dynamically and greedily rewired TRNs outperform original TRN subgraphs under all four robustness metrics under all conditions.

5.6.2. Simulation Experiments. In order to show the efficacy of the enhanced (i.e. rewired) TRN topology in real-world networks, we designed wireless sensor network (WSN) topologies based on original TRN subgraphs, rewired *E. coli* and Yeast TRN and E-R random graphs. We implemented WSN topology on OMNET++ Castalia simulator [114]. The simulations were performed on 50 subgraphs of 300 nodes. Each experiment was repeated 5 times. Following are the details of the simulation setup. (Additional simulation details are summarized in Table 5.4.)

Routing protocol: We follow the Collection Tree Protocol (CTP) [115], a tree-based distance vector routing protocol designed for sensor network communication.

Choice of Sink Nodes: The CTP protocol is a routing protocol where data is transferred from the source to the sink nodes, via a routing tree. CTP fits the TRN topologies since it becomes easy to model the hub nodes as sinks. To this end, the edge directions are reversed; the tier 3 nodes become the source nodes and the tier 1 nodes (with high incoming edges in the reversed graph) are the sink nodes. On an average, 5% hub nodes in each network are considered as sinks.

Table 5.4. Simulation Parameters.

Parameter	**Value**
Carrier frequency	2.4 GHz
Data transmission rate	250 Kbps
Transmission power level	0 dBm
Transmission, Reception, Sleep power	57.42mW, 62.0mW, 1.4mW
Receiver Sensitivity	-95.0 dBm
Initial energy of nodes	18720 J

Node deployment and communication: In order to generate WSN topologies based on the four topologies, let us consider any input subgraph (of original TRN, rewired TRN or E-R random graph) $G_g(V_g, E_g)$ and already deployed WSN topology $G_w(V_w, E_w)$, where the coordinate of node $u \in V_w$ is given by C_u. In course of this mapping procedure between G_g and G_w, we generate subgraph *mapped-WSN* G'_w ($V'_w \subset V_w$ and $E'_w \subset E_w$) and mapped-TRN G'_g ($V'_g \subset V_g$ and $E'_g \subset E_g$). Specifically, we define a simple greedy algorithm inspired from [45] to find a one-to-one mapping between a node pair $x \in E'_g$ and $w \in E'_w$ if for each $e(u, x) \in E'_g$ there exists $e(v, w) \in E'_w$, where u is already mapped to v. The details of the mapping algorithm has been covered in Section 3 of Appendix B. It returns a mapping function $\mathbf{m} : V_g \longrightarrow V_w$. Each node u in mapped-WSN G'_w is deployed in position C_u. Moreover, each mapped node in G'_w maintains a list of neighbor nodes to refrain from transmission or reception of data packets from non-neighbors in its vicinity.

In order to realize a TRN topology G_g (of 300 nodes), we consider a randomly deployed G_w of 400 nodes in an area of 100×100 sq. meters. An edge exists between each node pair of G_w if the they are within a range of 60 m. Given each graph G_g, we invoke above mapping algorithm to generate the mapped-WSN G'_w. In Table 5.5, we estimate the average percentage of nodes in G_g that are mapped to G_w. We observe that there is no notable improvement in percentage of mapped nodes when we have G_w in excess of 400 nodes, since some nodes with poor reachability in G_g remain unmapped.

Deployment area: The nodes are deployed on a simulation area of 100×100 sq. meters.

Simulation time: Each simulation is carried out for 1800 seconds.

Failure model: Like in Section 5.6.1, we consider three scenarios: (a) no node failure (b) random node failure and (c) targeted node failure. In case of targeted and random failures 4% nodes are removed from each topology after every 300 seconds.

Table 5.5. Average percentage of nodes G_g mapped for each topology type.

	E. coli	**Yeast**
Original	95.0%	96.5%
Greedy	95.9%	95.9%
Dynamic	94.4%	96.2%
Random	93.6%	94.6%

Metrics: We consider two metrics (a) *Average Packet Delivery Ratio (PDR):* It is the ratio between the number of packets received by all the sink nodes to the total number of packets generated by all the nodes and (b) *Average Latency-to-PDR ratio:* Latency is the time taken by a data packet to travel from source to a preassigned sink node. However, we observe that network latency alone is not adequate to gauge communication delay, as it is often negligible when majority of data are not forwarded due to unavailability of source-destination paths. Therefore, instead of considering average latency, we evaluate latency-to-PDR ratio for our experiments.

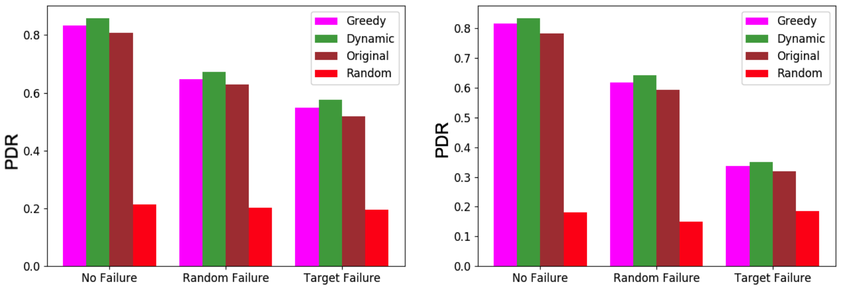


Figure 5.17. Packet delivery ratio. (a) *E. coli* WSN (b) Yeast WSN.

5.6.2.1. PDR. Figure 5.17(a) and 5.17(b) show that dynamic TRNs exhibit the highest PDR, while PDR for greedily rewired TRN is higher than original TRN, E-R graphs.

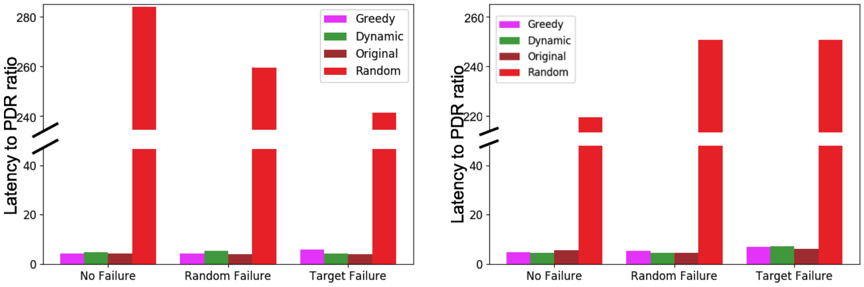


Figure 5.18. Network latency. (a) *E. coli* WSN (b) Yeast WSN.

5.6.2.2. Latency-to-PDR ratio. Figure 5.18(a) and 5.17(b) show that the Latency-to-PDR ratio of original, greedy and dynamic *E. coli* and Yeast TRN are comparable, whereas E-R random graphs possess a very high latency-to-PDR ratio.

We infer that the WSN simulation results corroborate our graph robustness experiments. The rewired TRNs exhibit a notably higher PDR at comparable latency, under both random and targeted failure conditions. (Another perspective behind the observed WSN performance in rewired TRNs has been discussed in Section 4 of Appendix B.) Thus, rewired TRNs are indeed an effective choice for design of robust and efficient WSN topologies.

5.7. INFERENCES

In this section we analyzed the topologies of both *E. coli* and Yeast TRN in terms of the three tier topology. We discussed that *E. coli* and Yeast TRN are vulnerable to the failure of well connected nodes. To remedy this vulnerability, we introduced an edge rewiring mechanism, consisting of edge addition and edge deletion. We discussed the edge addition and then the greedy edge deletion algorithm with its drawbacks. We then formulated the dynamic edge deletion algorithm as a nonlinear least squares optimization problem. We showed that edge rewiring preserves the key graph properties such as scale free out-degree distribution, motif abundance and graph sparseness. We then carried out through graph and simulation experiments to show that dynamically rewired TRNs not only exhibit the highest FFL motif count, but also outperform greedily rewired TRN, original TRN and E-R random graphs in terms of four robustness metrics, namely network efficiency, preservation of short source-sink paths and preservation of largest connected components, in events of random and targeted node failures. Finally, we demonstrated that rewired TRNs can be applied to real-world communication networks by performing WSN simulations on OMNET++, where dynamically and greedily rewired TRN-based WSNs exhibited the highest packet delivery at comparable network latency compared to other topologies.

6. A BIO-INSPIRED PROBABILISTIC DATA COLLECTION FRAMEWORK FOR PRIORITY-BASED EVENT REPORTING IN IOT ENVIRONMENTS

Over the last decade, unprecedented rise in population and unplanned land usage has led to the lack of sustainability in urban environments. Smart cities aim at using *Information and Communication Technology (ICT)* to develop energy-efficient applications, augmented with automated decision making, to support various public services in urban spaces. The paradigm of the *Internet-of-Things (IoT)* is considered to be a key enabler of such ICT-based smart city applications. It is an interconnection of a wide array of devices with sensing, computing, and actuation capabilities. IoT also creates a sense of pervasive computing by enabling devices to communicate with other devices and users via smartphones and wearables, as well as with application platforms hosted in cloud via backbone networks.

Energy-efficient smart-city applications require the energy-constrained IoT devices to operate over long durations without compromising the quality of sensing, processing, and transferring the collected data [131], implying that the lifetime of IoT devices are critical. The devices are often deployed at remote locations and their limited energy gets dissipated while sensing the environment and communicating the sampled data via wireless communication technologies, such as 3G/4G/LTE, WiFi, ZigBee, or Bluetooth/BLE [64]. It may not be always feasible to replenish their batteries or replace them with fully-charged devices on-the-fly. Quality of reported data is also essential, since the actuation which the application platform triggers (based on sporadically received information) will be erroneous and unproductive to the end users.

Clearly, sensing and reporting environmental events are important tasks of IoT-based data collection framework. In this paper, we leverage TRNs to design energy-efficient and QoI-aware data collection framework, called *bioSmartSense*+, for smart city applications based on self and neighbor regulation mechanisms in TRNs. We formulate an optimization problem to select a subset of IoT devices for sensing and reporting tasks, by satisfying energy-efficiency and QoI requirements. After proving NP-completeness of the optimization problem, we propose a sub-optimal algorithm by customizing a heuristic for the *Maximum Weighted Independent Set (MWIS)* problem. Let us look at the additional features of the proposed framework.

1. We consider a more realistic scenario where an event has prespecified priority. This enables the IoT devices (with limited residual energy) to conserve energy by preferentially reporting the high priority events.

2. We present a realistic probabilistic sensing model and comprehensively studies the effects this model has on the performance of the framework.

3. We consider a network of heterogeneous IoT devices with varying event sensing capacities and energy consumption rates. We model this diversity by designing varying device deployment profiles and study its effect on average residual energy and event reporting.

4. We validate by generating events from a real data set on traffic alerts.

5. Given a constrained system energy budget, we experimentally demonstrate how *bioSmartSense+* enhances network lifetime over the preliminary *bioSmartSense* framework using both synthetic as well as real event distributions.

6.1. BIO-INSPIRED SYSTEM MODEL

In this section, we describe the system model for the proposed framework.

6.1.1. Transcriptional Regulatory Networks. We represent a TRN as a directed graph $\mathcal{G}_g(\mathcal{V}_g, \mathcal{E}_g)$ where $\mathcal{V}_g$ are TFs/genes and $\mathcal{E}_g$ are regulatory interactions between TFs and genes. More details on the graph theoretic properties of TRN can be found in [95, 132, 133]. Broadly, there are four possible types of regulation in TRN:

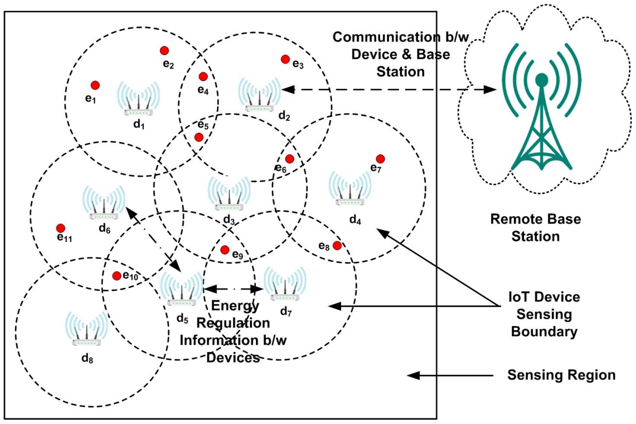


Figure 6.1. Bio-inspired system model.

1. *Activation (+)*: Increase in concentration of the TF increases the gene's concentration.

2. *Repression (−)*: Increase in concentration of TF decreases gene's concentration.

3. *Dual (+−)*: The regulating entity may activate or inhibit concentration of target gene.

4. *Self-regulation*: The regulating entity may activate or inhibit its own gene activity.

6.1.2. System Components. This work pertains to a network of IoT devices in smart city applications. We partition an urban area in a smart city into several grids, each of which is equipped with IoT devices $D = \{d_1, d_2, d_3, \cdots\}$ for sensing events related to environment and traffic. Figure 6.1 captures one such grid. The events sensed by the IoT devices are reported to a remote application platform, called the *base station*.

We define *time epoch* as a temporal window at which vital processes take place in the system of IoT devices. Time epoch, defined as t $(= t_r + t_s + t_b + t_{tx})$, is partitioned into four steps which are as follows:

1. *Device energy regulation (t_r)*: Each device exchanges control messages (termed *regulatory messages*) and modulate their sensing intensity (termed *energy level*).

2. *Event sensing (t_s)*: Each device senses the different events taking place in its sensing range.

3. *Device-base station beacon exchange (t_b)*: Each device sends out beacon messages to the base station notifying it of the unique identification of the events sensed as well as energy level, based on which the latter decides whether the device should participate in event reporting.

4. *Information transmission (t_{tx})*: The devices chosen to report the events to the base station carry out transfer of complete event information.

We discuss the major components of our system as follows:

1. IoT device: Each device d_i is an energy-constrained node capable of probabilistically sensing events in radius r_d (see Section 6.3.2.2), which are processed and conditionally reported to the base station using WiFi, Bluetooth low energy (BLE) or ZigBee protocols. Each IoT device interacts with others in its sensing range through control (or regulatory) messages. We term the set of IoT devices D together with its communication links, as the *IoTNet* topology. *IoTNet* is an undirected graph $\mathcal{G}_w(\mathcal{V}_w, \mathcal{E}_w)$, where $\mathcal{V}_w = d_1, d_2, ..., d_m$ represents the set of IoT devices and an edge $\{d_i, d_j\} \in \mathcal{E}_w$ exists if two devices d_i and d_j are within communication range of one another. Moreover, each device has an event sensing intensity (termed *energy level*), lying in range $[0, 1]$. We assume (in accordance with [134]) that at a higher energy level, a device can sense data at higher sampling rate leading to better sensing accuracy. However, operating at higher energy causes larger rate of dissipation of device's residual energy. IoT devices are capable of two types of energy regulations, called *neighbor regulation* and *self regulation* (details given in Section 6.2.2).

2. Event: Set of events is defined as $E = \{e_1, e_2, e_3, \cdots\}$. An event $e_k \in E$ (marked in red in Figure 6.1) is sensed by one or more IoT device(s). Given that the devices report traffic (or environmental) information, each event is assigned a priority score between 1 (low priority), for e.g., 'road closure', and 5 (high priority), for e.g., 'major accident'. Note that any two events in roughly the same location and same time epoch are considered one and the same.

3. Base Station: It is a central application platform that processes event reported by the IoT devices. It has a higher sensing range (r_B) compared to the IoT devices, and is capable of bidirectional communication.

6.2. PROPOSED *BIOSMARTSENSE*+ FRAMEWORK

In this section, we discuss the details of the proposed *bioSmartSense*+ framework.

6.2.1. IoTNet to TRN Mapping. As discussed in Section 3, TRN possesses few graph theoretic properties that render fault-tolerance and self-organization. *bioSmartSense*+ endeavors to exploit these attributes to develop an energy-efficient data collection framework. To this end, we map the *IoTNet* to TRN, using the mapping technique proposed in [44] [45]. It generates a subgraph of the *IoTNet* that has been proven to preserve the intrinsic graph robustness of TRN.

Given an *IoTNet* $\mathcal{G}_w(\mathcal{V}_w, \mathcal{E}_w)$ and a TRN graph $\mathcal{G}_g(\mathcal{V}_g, \mathcal{E}_g)$, the mapping algorithm yields a one-to-one node mapping function $M : \mathcal{G}_{mw} \rightarrow \mathcal{G}_g$ such that $\mathcal{G}_{mw}(\mathcal{V}_{mw}, \mathcal{E}_{mw})$ is a directed *mapped-IoTNet topology*, where $\mathcal{V}_{mw} \subset \mathcal{V}_w$ and $\mathcal{E}_{mw} \subset \mathcal{E}_w$. An edge $\{d_i, d_j\} \in \mathcal{E}_{mw}$ exists if and only if there is a path between $M(d_i)$ and $M(d_j)$ in $\mathcal{G}_g$. Illustrative example as well as the analysis of running time of the mapping algorithm has been covered in the preliminary version of this work [135].

6.2.2. TRN-based Energy Level Regulation. For any edge $\{u, v\}$ in a directed graph, u and v are termed the *predecessor* and *successor* nodes, respectively. Likewise, each IoT device d_i can possess a list of predecessor and successor nodes, given by $\phi(d_i)$ and $\eta(d_i)$, respectively. As mentioned earlier, we consider two kinds of TRN-based energy level regulations:

Neighbor regulation: At every regulation phase t_r, an IoT device d_i sends regulatory messages to $\eta(d_i)$ and receive regulatory messages from $\phi(d_i)$. Based on received messages, device d_i subsequently regulates its energy level ($l^t_{d_i}$) at time instance $t + 1$, with the following update rule: $l_{d_i}^{t+1} = \sum_{v \in \phi(d_i)} \kappa \times \mathbf{W}(d_j, d_i) \times l_{d_i}^{t}$. Here $\mathbf{W}$ is the edge weight i.e., $\mathbf{W} : \mathbf{W}(d_j, d_i) \longrightarrow \{+, -\}, \forall \{d_j, d_i\} \in \mathcal{G}_{mw}$ and κ is a rate constant that dictates the extent of positive or negative regulation of a d_i by its predecessors.

Self regulation: This is the other type of regulation where d_i monitors its own energy level l_{d_i}. The idea behind this is that if a d_i keeps high l_{d_i} over a prolonged time, it will consume high sensing energy; conversely, if it has low l_{d_i} over time, quality of events sensed will deteriorate. To find a balance, if l_{d_i} exceeds upper bound U_{th} or drops below lower bound L_{th} for a duration of over rI, it is reset to a baseline b_{th}.

6.2.3. Sub-optimal Selection of Sensing Devices. We discussed in Sections 6.1.2, *IoTNet* is an energy-constrained network. It is to be noted that a energy-constraint network can provider longer periods of uninterrupted service only if its residual energy is expended judiciously. In this work, we assume that the total permissible energy consumption of all the IoT devices is upper capped by $\mathcal{B}$. Thus, we define network lifetime *as the time instance when the collective energy consumed by sensing and reporting of all devices exceeds the predefined energy budget $\mathcal{B}$.*

In the context of smart city application, network longevity alone is not enough. It is imperative to ensure quality of service (QoS) of the *IoTNet*. Thus, the proposed *bioSmartSense+* framework aims at maximizing the quality as well as quantity of events sensed and reported at every time epoch. Specifically, given an event set $E = \{e_1, e_2, \cdots, e_n\}$, a set of devices $D = \{d_1, d_2, \cdots, d_m\}$ and an overall residual energy budget $\mathcal{E}^t = \sum_{i=1}^{m} \varepsilon_{d_i}^t$ at time epoch t, where $\varepsilon_{d_i}^t$ is the residual energy for device d_i, *bioSmartSense+* conserves energy by refraining from prompting all the IoT devices to report events to the base station. Instead, a near-optimal subset of IoT devices are intelligently chosen to report their events. The chosen devices satisfy two conditions: (i) *Quality:* its energy-level is higher enough to guarantee superior accuracy of reported events, and (ii) *Quantity:* it has sensed a high number of events during the last time epoch.

We term this as the problem of *Device Selection for Quality and Uninterrupted Information Dissemination (DSQUID)*. It is formally defined as follows:

(DSQUID). Select non-redundant event information from a set of devices, such that both the quality and quantity of the disseminated information are maximized, subject to constraint on the overall network residual energy budget.

6.2.3.1. NP-Completeness of DSQUID. As mentioned above, E is the set of events whose information will be used in the present time epoch to select reporting devices. Let $\mathcal{D}$ be a collection of device contributions $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_m$, where $\mathcal{D}_i$ comprises the events sensed by device d_i in the present time epoch.

Without loss of generality, we can simplify the $DSQUID$ problem by assuming $\mathcal{D}^* \subseteq \mathcal{D}$, such that **(1)** each event in E is contained in *at most* one subset in $\mathcal{D}^*$, and **(2)** each event in E is contained in *at least* one subset in $\mathcal{D}^*$. Condition (1) ascertains that no two devices who have sensed the *same* event are selected, while condition (2) ensures that information for *all* events sensed in the current time epoch are reported. Thus, the problem reduces to finding $\mathcal{D}^*$.

In [135], we have shown that there exists no polynomial time solution that finds the set of device contributions exactly covering the set of events. In essence, the $DSQUID$ problem is *NP-complete*. To prove this, we reduce a classic *NP-complete* problem, known as the *Exact Cover* [136], to the $DSQUID$ problem.

(Exact Cover Problem). Given a collection $\mathcal{S}$ of subsets of a set X, an exact cover of X is a sub-collection $\mathcal{S}^*$ of $\mathcal{S}$ that satisfies two conditions:

- The intersection of any two distinct subsets in $\mathcal{S}^*$ is empty, i.e., the subsets in $\mathcal{S}^*$ are pairwise disjoint. Thus, if $\mathcal{S}_i, \mathcal{S}_j \in \mathcal{S}^*$, then $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$.
- The union of the subsets in $\mathcal{S}^*$ is X, i.e., the subsets in $\mathcal{S}^*$ cover X. Thus, $\forall \mathcal{S}_i \in \mathcal{S}^*$, $\bigcup_i \mathcal{S}_i = X$.

We do the following construction. Let each $x_i \in X$ correspond to an event $e_i \in E$. Then $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_m\}$ maps to $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_m\}$, such that $\mathcal{S}_i$ corresponds to $\mathcal{D}_i$. The mapping can be shown in polynomial time. Thus, the set $\mathcal{D}^*$ discussed earlier now corresponds to $\mathcal{S}^*$. Hence, if we can find $\mathcal{D}^*$ in polynomial time, we can also solve the *Exact Cover* problem in polynomial time. As the *DSQUID* problem is *NP-complete*, at best we can find a sub-optimal solution.

6.2.3.2. Heuristic to solve DSQUID problem. The heuristic for finding a sub-optimal solution for the *DSQUID* problem is motivated by the approximation algorithm [137] proposed for the *Maximum Weighted Independent Set (MWIS)* problem. It has been used to find the maximal weighted independent sets in the topology graphs of wireless networks [138].

Let $G = (V, A, \omega)$ be a simple weighted undirected graph, where V is the set of vertices, A is the set of edges, and ω is the vertex weighting function such that $\omega : V \mapsto \Re^+$, $\omega(u) \in \Re^+$ for all $u \in V$, $\omega(S) = \sum_{u \in S} \omega(u)$ for any nonempty set $S \subseteq V$ and the set of positive reals $\Re^+$. A subset $I \subseteq V$ is an independent set of G if for any two vertices $u, v \in I$, $\{u, v\} \notin A$. An independent set I of G is maximum if there is no independent set I' of G such that $\omega(I) < \omega(I')$.

In the *DSQUID* problem, our objective is to choose a minimal set of devices which has collected better quality information (QoI) for maximum number of events, while ensuring non-redundancy of events reported. Such choice enables dissemination of information of nearly all events occurred in the current time epoch with higher degree of accuracy, subject to the constraint that the energy dissipated does not exceed an overall residual energy budget. The weighing (fitness) function for device selection needs to incorporate both *quantity* (i.e. sensing of different events) and *quality* (i.e. accuracy in generated report) of contribution.

For the *MWIS* problem, Sakai *et al.* [137] proposed a generalized weighted greedy algorithm, called $GWMAX$. It is an extension of the existing $GMAX$ algorithm that selects a vertex of maximum degree, removes it and its neighbors from the graph, and iterates this process on the remaining graph until no vertex is remaining, implying the set of selected vertices is an independent set. $GWMAX$ has an approximation ratio of at least $\frac{1}{\Delta}$, where Δ denotes the maximum degree of the given graph $G(V, A, \omega)$. It generalizes the vertex selection rule as: *Select each* v_i $(0 \leq i \leq |I| - 1)$ *that satisfies*

$$\sum_{u \in N_{G_i}(v_i)} \frac{\omega(u)}{deg(u)(deg(u)+1)} \geq \frac{\omega(v_i)}{deg(v_i)+1} \tag{6.1}$$

where, $N_{G_i}(v_i)$ is the set of vertices adjacent to v_i in the subgraph G_i and $deg(v_i)$ is the degree of vertex v_i. This implies that the vertex weight normalized by its degree plus 1 forms the selection criteria for all the vertices, and the one which has maximum normalized weight gets selected.

In the beginning of the $GWMAX$ algorithm, a set I is initialized to be an empty set, and the weight of each node in G is evaluated with Eq. (6.1). Then, the $GWMAX$ iteratively selects a node v_i using maximum value in G and adds v_i to I, until no node can be selected. In each iteration, when a node v_i with maximum value in G is selected, G is updated as a subgraph of G_i induced by $V - N_{G_i}(v_i)$. In addition, reevaluation of the weights in the current subgraph is carried out using Eq. (6.1). We will use a variant of the $GWMAX$ heuristic as the device selection criteria in the $DSQUID$ problem.

6.2.4. DSQUID Algorithm. Consider an event set $E = \{e_1, e_2, ..., e_n\}$ and device set $D = \{d_1, d_2, ..., d_m\}$. We undertake a graph transformation to create an undirected weighted graph $G(D, A, \omega')$, where the set of vertices denote the D and an edge $\{d_i, d_j\}$ exists if both d_i and d_j have sensed the same event. If a device d_i has degree k in $G(D, A, \omega')$, it implies that the d_i has sensed the same event(s) as k other devices. The vertex weight ω' (discussed below) is a weighing function indicative of the quality and quantity of events sensed by a device. The primary purpose of this graph transformation is to identify unique devices for event reporting that have the least overlap in events sensed. In Figure 6.2, we show the graph transformation of the *IoTNet* (shown in Figure 6.1), which serves as an instance of the *DSQUID* problem at a certain time epoch. We run the *GWMAX* algorithm on G to identify the subset of nodes selected for event reporting.

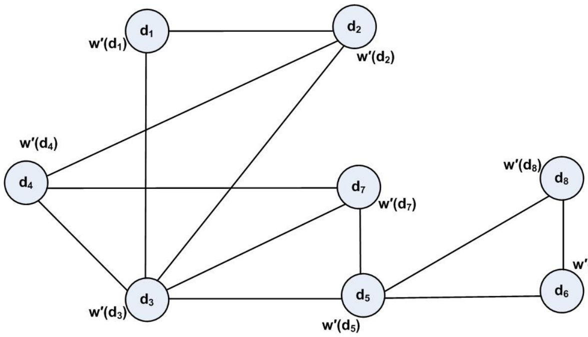


Figure 6.2. Graphical transformation of the *DSQUID* problem.

6.2.4.1. Vertex weighing function. The *bioSmartSense+* framework strives to maximize both the quality as well as quantity of the events reported to the base station. At time t, for each device d_i in the transformed graph G, we define $\omega'(d_i)$ as a weighted function of the quality of events sensed (given by QoI index $q^t_{d_i}$) and quantity ($n^t_{d_i}$), as:

$$\omega'(d_i) = \sigma \cdot n^t_{d_i} + (1 - \sigma) \cdot q^t_{d_i} \tag{6.2}$$

Here σ (s.t. $0 \leq \sigma \leq 1$) denotes the *preference factor* which controls the importance attached to quality and quantity. Note that σ depends on contextual information, such as the location, temporal biases, etc. If the *IoTNet* environment needs to tackle higher event occurrence rate, σ can be set to a larger value. Conversely, if the event occurrence is sporadic, a smaller σ is the preferred choice. Let us define the expected QoI index.

Recall that we expect that the quality of the data samples collected to be contingent on the instantaneous energy level of any device d_i ($l^t_{d_i}$) lying in range [0, 1]. To this end, we apply the following non-linear relationship [134] to quantify QoI index:

$$q^t_{d_i} = \alpha \cdot (l^t_{d_i})^{\beta} \tag{6.3}$$

Here α ($0 < \alpha < 1$) is the attainable QoI index, given that the energy level $l^t_{d_i}$ is maximum; β ($0 < \beta < 1$) is the discounting factor. Both α and β are subject to variation depending on application as well as the spatial and temporal aspects of the sensing environment.

6.2.4.2. Energy consumed by IoT devices. Let us denote the devices selecting for event reporting as D^* ($D^* \subset D$). At each epoch, any $d_i \in D^*$ needs to perform both sensing and reporting tasks, making its total energy dissipation:

$$\epsilon^t_{d_i \in D^*} = l^t_{d_i \in D^*} \cdot (\delta_1 + n^t_{d_i \in D^*} \cdot \delta_2) + \delta_3 \tag{6.4}$$

Here, δ_1 is a constant measure of energy expended by d_i in idle mode and sensing, δ_2 is the constant energy to transmit one event, and δ_3 is the fixed energy to activate transmitter radio.

The devices not selected for event reporting ($D \setminus D^*$) need not activate radio nor transmit and their total energy dissipation in time epoch t is given by:

$$\epsilon^t_{d_j \in D \setminus D^*} = l^t_{d_j \in D \setminus D^*} \cdot \delta_1 \tag{6.5}$$

The application of the $GWMAX$ algorithm solution on the $DSQUID$ over a total time epoch T solves the following optimization:

$$\begin{aligned} &\text{maximize} && \sum_{i=1}^{|\mathcal{D}^*|} \omega'(d_i) \\ &\text{subject to} && \sum_{t=1}^{T} \left(\sum_{i=1}^{|D^*|} \epsilon^t_{d_i \in D^*} + \sum_{j=1}^{|D \setminus D^*|} \epsilon^t_{d_j \in D \setminus D^*} \right) \le \mathcal{B} \end{aligned} \tag{6.6}$$

In summary, the $DSQUID$ problem essentially creates a schedule of devices for event reporting, with the objective to maximize quality and quantity of events reported, while ensuring the overall energy consumed is confined within a budget $\mathcal{B}$. The steps involved are summarized below:

1. Generate a transformed graph $G(D, A, \omega')$ from an instance of the $DSQUID$ problem.
2. Utilize the $GWMAX$ algorithm to find the independent set I (schedule of devices) for event reporting.

6.2.5. Communication Protocol. The communication protocol dictates how messages are exchanged (1) among IoT devices and (2) between the base station and IoT devices. At the *energy regulation phase*, each device carries out its TRN-based energy regulation. Given r (short for *regulatory*) and t is the present time epoch, a device sends a message of format ⟨r, $device\_ID$, $energy\_level$,t⟩ to its successor devices and regulates its own energy level (see Section 6.2.2). Moreover, in the IoT device-base station beacon exchange phase, there are three steps:

Step-1: A device sends out a beacon message to the base station listing the unique IDs of events sensed. The format of the beacon is $\langle sb, device\_ID, energy\_level, [event\_IDs] \rangle$, where the type of message is sb (short for *sensing device-to-base station*).

Step-2: The base station uses Eqns. (6.2) and (6.3) to find the $\omega'(d_i)$ for every IoT. The IoT devices are chosen to report the events to the base station using $DSQUID$ algorithm. Following this, the chosen IoT devices receive beacons asking them to send the complete event information to the base station.

Step-3: The selected devices send the complete event information to the base station.

6.3. AUGMENTED CAPABILITIES IN *BIOSMARTSENSE*+ FRAMEWORK

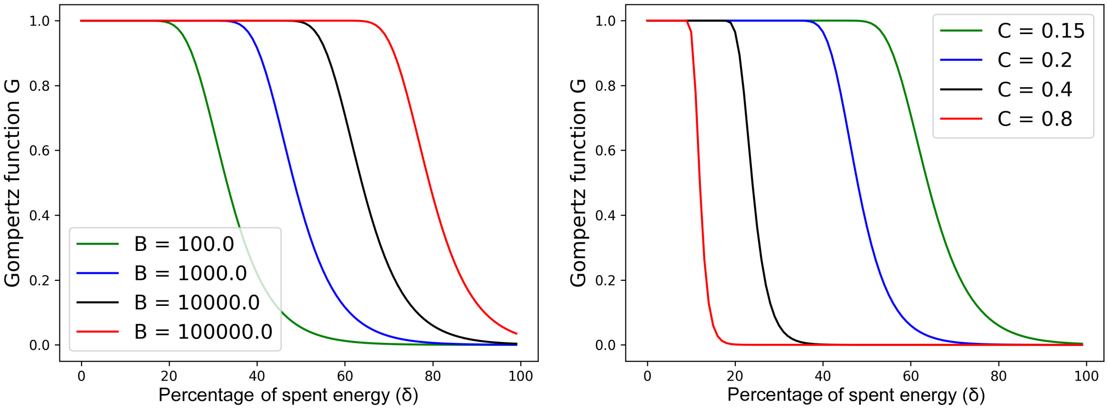
In this section, we discuss the new capabilities that have been included in the *bioSmartSense*+ framework to conceive realistic sensing and reporting mechanisms under heterogeneous device deployment.

6.3.1. Event Reporting Based on Priority. The events which occur in a smart city setting can have varying priorities. For instance, a traffic event pertaining to a potential road hazard due to pothole has a low priority, while a major road accident will have a high priority. The IoT devices chosen for reporting the events to the base station are drained of data transmitting energy. Instead of reporting all the sensed events, we intuit, the energy-depleted devices can be made to report the K highest priority events only to conserve energy. Thus, attaching a priority score to an event may help these devices to carry out preferential reporting and prevent unnecessary energy dissipation.

We assign a priority score to each event on a scale of 1-5, where 1 is the lowest and 5 is the highest priority score. Each IoT device sorts the sensed events in the decreasing order of priority score. At time epoch $t = 0$, we define energy g, as the percentage of spent energy equals 0 and it can potentially report all sensed events. Over time, as g $(0 \leq g \leq 1.0)$ increases, the device performs preferential reporting top-priority events. Since there is an inherent non-linearity in preferential data reporting mechanism, the *Inverse Gompertz (IG)* function [139] has been used for modeling preferential reporting:

$$G(A, B, C, g) = 1 - A \cdot e^{-B \cdot e^{-C \cdot g}} \tag{6.7}$$

In Eq. (6.7), $G(A, B, C, g)$ is the fraction of top priority events reported and g is the percentage of spent energy, both pertaining to a particular IoT device d_i, which has sensed E_s^i events in the last time epoch. The Gompertz parameter B controls the length of time for which the curve steadily maintains its highest value, C determines the rate of decay of the curve, and A is the upper asymptote which is fixed at 1. Thus, in absolute terms, device d_i reports $K(= *G(A, B, C, g) * |E_s^i|)$ highest priority events.



(a) Displacement Parameter: B

(b) Decay Parameter: C

Figure 6.3. Variation of Inverse Gompertz parameters.

Figure 6.3(a) illustrates the effect of parameter B (of Eq. (6.7)) on the initial value of the fraction of top-priority events reported, before the latter enters into the exponential decay phase. For a fixed decay rate (considered to be $C = 0.2$), this parameter acts as a discounting factor to the highest proportion of events reported. The choice of its value depends upon whether any location or at a given temporal window, the probability of occurrence of top priority events is high. Higher value of B enables reporting of majority of such events. In contrast, if the likelihood is low, it can be set to a smaller value.

Parameter C controls the rate of decay in the fraction of top priority events reported, lower bounded by 0, as shown in Figure 6.3(b). We considered $B = 1000$ to study the varying effects of the growth parameter. For higher values of C, the reporting rate drops rapidly with relatively lower percentage of spent energy. Thus, if any location or time slot has higher likelihood of occurrence of top priority events, the *IoTNet* administrator can set a lower C to collect reports of majority of the events at the expense of device energy. However, if the likelihood is on the lower side, then a higher C will drop the reporting rate sharply and prevent further dissipation of energy.

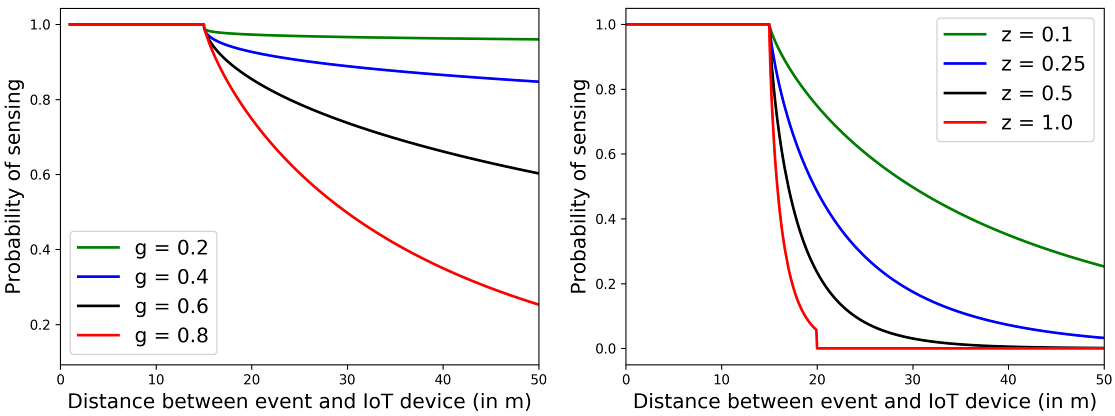
6.3.2. Sensing Model. In our preliminary data collection framework, *bioSmart-Sense* [135], we assume that the IoT devices follow the *boolean sensing model*. Here, we study and incorporate another sensing model, called the *probabilistic sensing model*, to conceive a more realistic sensing mechanism and at the same time achieve comparable energy efficiency and event reporting rate. The following subsections delineate the two sensing models.

6.3.2.1. Boolean sensing model. Boolean (deterministic) sensing model is the simplest and most commonly used sensing model [140]. In this model, if an event in the network field is located within the sensing range R of sensor node S, then it is assumed that it is covered/detected by the sensor S. The sensing area of S is defined by a circumference of radius R, centered at its location. Formally, boolean sensing model is defined as follows:

$$C(d) = \begin{cases} 1, & \text{if } r_d \leq R \\ 0, & \text{Otherwise} \end{cases} \tag{6.8}$$

Where, $C(r_d)$ is the coverage probability of an event that has occurred at an euclidean distance of r_d units from the location of the sensor. This model does not incorporate the environmental setting such as obstacles in the vicinity or the strength of the emitted signal on the task of sensing [141].

6.3.2.2. Probabilistic sensing model. The probabilistic sensing model [142] is a realistic extension of the boolean sensing model. This model is motivated by the fact that the quality of sensing gradually decreases with increasing distance from the sensor. Therefore, the coverage function needs to be expressed in probabilistic terms. In this work, we use the *Elfes sensing model* [143] to quantify the sensing probability.



(a) Sensing Radius Depletion Rate

(b) Fraction of Energy Spent

Figure 6.4. Parameters of the Elfes sensing model.

According to this model, the probability that a sensor detects an event to a distance r_d is given as:

$$C(r_d) = \begin{cases} 1, & \text{if } r_d < R_{min} \\ e^{-z(d-R_{min})^g}, & \text{if } R_{min} \leq r_d < R_{max} \\ 0, & \text{if } r_d \geq R_{max} \end{cases} \tag{6.9}$$

where, R_{min} defines the sensing radius within which all events are sensed, R_{max} is the sensing range of the device in excess of which no events are sensed. Note that when $R_{min} = R_{max}$, this model is reduced to the boolean sensing model. Parameters z and g are adjusted according to the physical properties of the sensor. We assume that device heterogeneity depends on these two parameters. Intuitively, we term these parameters as the sensing radius depletion rate and the fraction of energy spent till the last time epoch, respectively.

In Figure 6.4(a), we study the effect of parameter z on the sensing probability for a fixed value of parameter $g = 0.5$. It is evident that for lower sensing radius depletion rate, the device can detect larger proportion of events even if the distance r_d is on the higher side. Conversely, higher z causes rapid decay in the detection probability for relatively lesser distance.

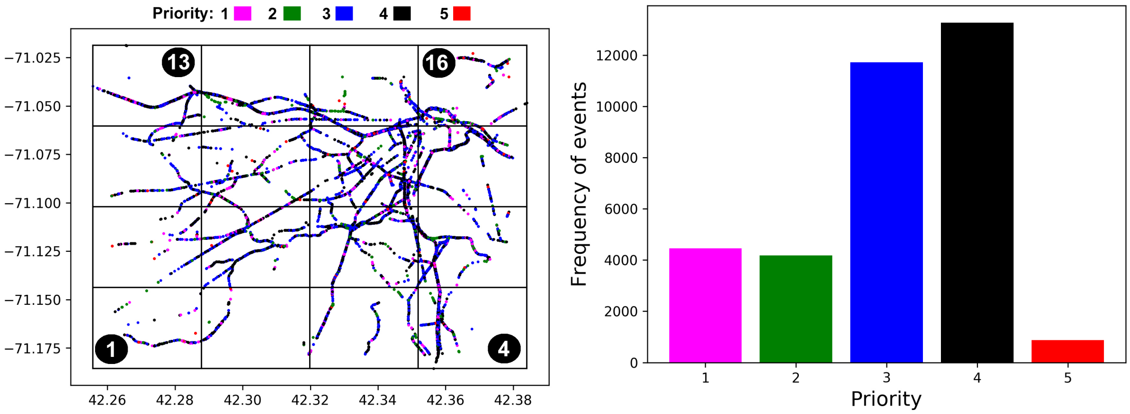
Figure 6.4(b) shows the variations of the sensing probability under different fractions of spent energy g. It is to be noted that for a fixed depletion rate $z = 0.7$, the detection probability drops sharply if the device battery is completely drained out ($g = 1.0$). On contrary, the detection probability slowly amortizes over greater distances if the device has spent only 10-25% of its residual energy.

6.4. EXPERIMENTAL RESULTS

We develop a customized discrete event simulator based on Python Simpy library [144]. We validate the proposed framework through experiments on synthetic as well as real data detailed in Sections 6.4.1.1 and 6.4.1.2, respectively. We compare the results with a state-of-the-art data collection framework proposed from the perspective of smart city applications [64], which, to the best of our knowledge, is the only work that aligns with our proposed data collection framework.

Table 6.1. Simulation parameters for *bioSmartSense*.

Parameter	**Symbol**	**Value (default)**
No. of IoT devices	N	100
Deployment region	-	2×2 sq. km
Energy budget	B_e	$6 \times 10^5 J$
Initial energy of IoT device	E	$15 \times 10^3 J$
Device memory	M	500 MB.
Simulation duration	T	100 min.
Self regulation energy level thresholds	L_{th}, U_{th}, b_{th}	0.2, 0.8, 0.6
Random event generator mean	m	40
Dist. btwn. event and IoT device	r_d	–
QoI preference factor	σ	0.5
QoI parameters	α, β	0.8, 0.8
Min. and max. event sensing radius	R_{min},R_{max}	30, 100 m
Inverse Gompertz parameters	A, B, C	1.0, 1000.0, 0.2
Coverage function parameters	z, g	0.7, 0.5
Packet reception, sensing energy	e_t, e_s	0.011, 3.68 J



(a) Spatial Event Distribution

(b) Priority-wise Event Frequency

Figure 6.5. Boston street data acquired from Waze.

6.4.1. Simulation Setting and Parameters. We create a deployment area of 2 × 2 sq. km, where 100 IoT devices, equipped with WiFi connectivity, are deployed at random points. The minimum and maximum event sensing radii of IoT devices, R_{min} and R_{max} are 30 and 100 meters, respectively. Each IoT device d_i possesses rechargeable battery energy capacity $\varepsilon_{d_i} = 15 \times 10^3$ J. Note that the cumulative energy budget $\mathcal{B} = 6 \times 10^5$ J, in excess of which operations of sensing and reporting tasks are stopped. In our experiments, unless otherwise stated, all parameters follow the default value summarized in Table 6.1.

6.4.1.1. Events generated using random data. In this case, events occur at random locations within the deployment region and their frequency in each epoch follows an exponential distribution $X \sim exp(m)$, where m is the mean. For every event we construct a boundary with a fixed radius of r meters. If an event is sensed by two or more devices whose locations are within the former's boundary, then we assign the same event identifier to the latter. Each experiment has been done on 50 different simulated data instances.

6.4.1.2. Event generation using real data. We generate another set of events using a week's real data on traffic related events spanning over 100 boroughs (names withheld in the interest of space) in Boston, MA, USA. This data has been shared by Google's Waze (www.waze.com) and has been used in our previous works [145] [146]. However, as the data is currently unavailable in its original source, we share a part of it in the public domain (https://github.com/satunr/StreetData.git). The details of the data have been summarized in Table 6.2.

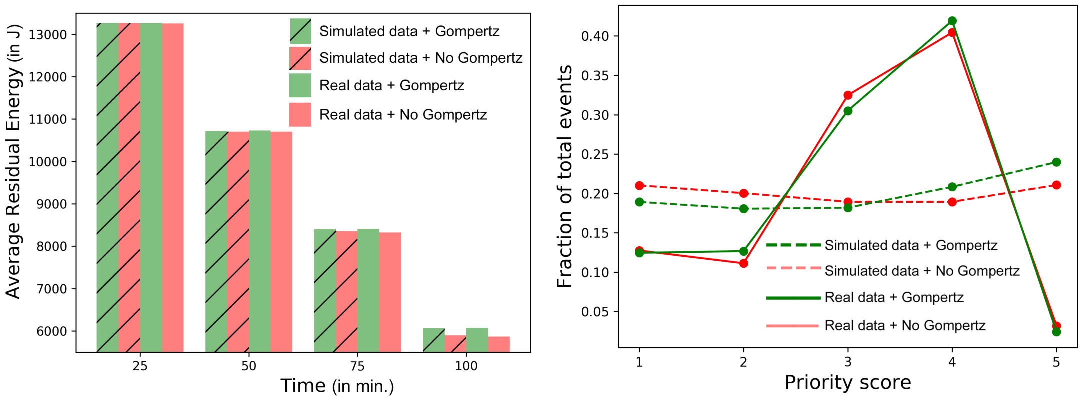
Table 6.2. Details of Waze data.

Field	**Value**
Number of events	$34,490$
Event priority	1 (low) - 5 (high)
Types of events (with priority)	hazard on road(3), accident major (5), road closed construction (1), road closed hazard (2), hazard on road car stopped (2), jam stand still traffic (3), hazard on road object (3), accident minor (5), hazard on shoulder car stopped (1), hazard on road pot hole (2), road closed event (1), hazard on road construction (1), major hazard on road (5), jam moderate traffic (3), hazard weather freezing rain (5), hazard on shoulder missing sign (1), hazard on shoulder animals (5), jam heavy traffic (4), hazard on shoulder (5), hazard weather (5), hazard on road ice (5), hazard weather fog (5)
Number of boroughs	102
Latitudinal extent	42.26 to 42.38
Longitudinal extent	-71.175 to -71.025

We divide the Boston region into 4×4 spatial grids of equal sizes. Figure 6.5(a) shows the event distribution of the data set, where different colors signify the priority of the events in the region. Note that the grids are named (in black circles) row-wise from bottom left to top right and X and Y axis correspond to latitude and longitudes, respectively. Figure 6.5(b) shows the frequency distribution of events of different priority.

For each grid x in the deployment region, we maintain an ordered 5-tuple $(e^x_{p_1}, e^x_{p_2}, \cdots, e^x_{p_5})$, where $e^x_{p_i}$ is the frequency of events with priority i. Here also we use exponential distribution $X \sim exp(m)$ to model the number of events generated over time. In each time epoch t, the grid location and priority type of $X(t)$ events are generated. While generating each event, the following steps are considered:

1. Select a random grid x with a probability commensurate with the number of events that took place in the grid as per the Waze data i.e., $p_x = \frac{\sum_{i=1}^{5} e^x_{p_i}}{\sum_{x=1}^{16} \sum_{i=1}^{5} e^x_{p_i}}$
2. In grid x, select an event of priority type i with a probability commensurate with the frequency of events of type i that took place in the selected grid x i.e., $p_i = \frac{e_{p_i^x}}{\sum_{j=1}^{5} e^x_{p_j}}$



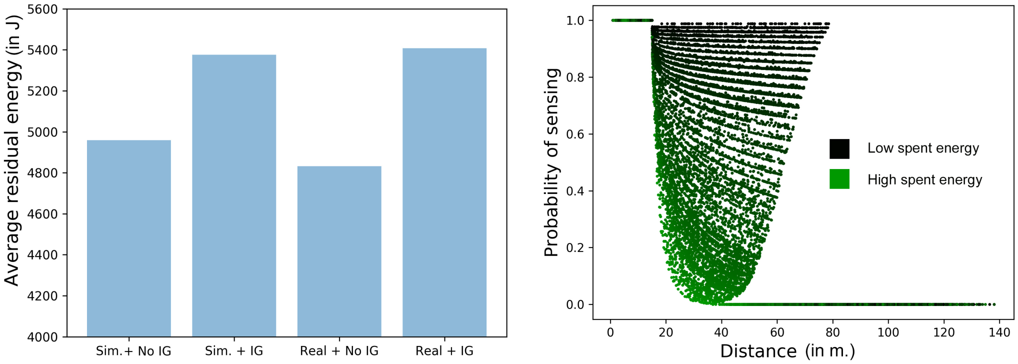
(a) Average Residual energy

(b) Fraction of Events Sensed

Figure 6.6. Effect of preferential event reporting.

6.4.2. Effect of Preferential Event Reporting. The Inverse Gompertz (IG) function allows the IoT device to select the top K highest priority events to be reported. As a consequence, *bioSmartSense*+ conserves energy, while preferentially reporting the highest priority events to the base station.

Figures 6.6(a) and 6.6(b) show that for both cases of simulated as well as real data, the use of the Inverse Gompertz (IG) function leads to a marginally improved overall energy efficiency while reporting a higher fraction of high priority events. Note that the devices whose residual energy are conserved by employing the IG function are those that are selected to report events to the base station. The devices selected for event reporting only constitute a small fraction of total IoT devices in the system, therefore the improvement in the overall residual energy of the system is marginal (see Figure 6.6(a)). However, when we consider the mean residual energy of devices selected for reporting devices, the improvement is significant (refer Figure 6.7(a)). It shows that the IG function causes a notable improvement in the residual energy for the devices reporting events generated from simulated and real data.



(a) Effect of IG on Reporting Devices' Energy

(b) Effects of Dist. and Spent Energy on sensing

Figure 6.7. Augmented capabilities of *bioSmartSense+*.

6.4.3. Effect of Probabilistic Sensing on Event Coverage. Given the distance between the location of event and IoT device (r_d), the probability of event sensing $C(d)$ (Eq.(6.9)) is affected by the fraction of spent energy g as wen as r_d. We now analyze the variation in $C(d)$ for different r_d and g. Given $z = 0.5$, Figure 6.7(b) shows two scenarios. With the increase in r_d, if the value of g is low (shown in dark green), the drop in C_d with negligible; conversely, if g is high (shown in light green), the decline in C_d is also high.

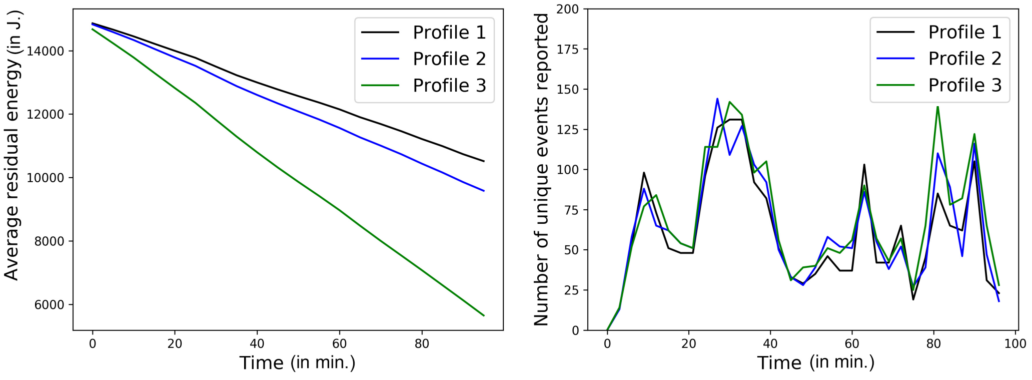
6.4.4. Effect of Device Heterogeneity on Event Reporting. In [135], we assumed that the IoT devices possess identical configuration w.r.t. memory as well as energy consumption rate. We consider a more realistic setting where IoT devices are heterogeneous and have varying system configurations in terms of device memory (M), energy consumption rate for sensing (δ_1) and sensing radius depletion rate (z). To realize device heterogeneity, we consider three device types, details of which are depicted in Table 6.3.

Now, we take three deployment profiles containing different proportions of Type-1, 2, and 3 devices. Let *Profile-1* have (10%, 10%, 80%), *Profile-2* have (10%, 80%, 10%), and *Profile-3* have (80%, 10%, 10%) IoT devices of three types. Therefore, *Profile-1*, which is dominated by low efficiency devices, is expected to show lowest event sensing and reporting at the expense of low sensing energy consumption. Analogously, the other two profiles have medium and high event sensing and reporting potential, respectively.

Table 6.3. Configuration of Heterogeneous Devices.

	Rate (z)	Sensing energy (δ_1)	Memory (M)
Type-1	0.05	3.68 J	500 MB
Type-2	0.50	1.40 J	100 MB
Type-3	1.0	0.70 J	50 MB

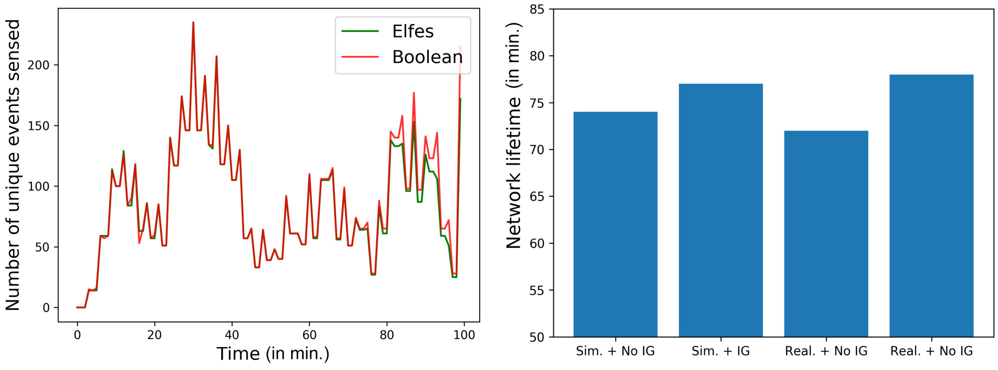
Figures 6.8(a) and 6.8(b) show the average residual energy and number of unique events reported over time, under event prioritization scheme. As expected, the plots depict that the *Profile-1* which possesses the maximum number of low efficiency devices consumes least energy. However, due to redundancy in event sensing, the difference in event reporting by *Profile-3* is only marginally higher than *Profile-1*. We performed the same experiment without IG event prioritization. The results are similar (not shown here), though the average residual energy are marginally higher and number of events reported is marginally lower as expected.



(a) Average Residual Energy (b) Number of Unique Events Reported

Figure 6.8. Results on device heterogeneity.

6.4.5. Effect of Probabilistic Sensing. The probabilistic or Elfes sensing model is more realistic in capturing the event sensing potential of a IoT device. Recall that, unlike the boolean model, in Elfes, the events occurring at a distance in the range $[R_{min}, R_{max}]$ are only sensed with a probability (as defined by Eq.(6.9)). Thus, we intuit that probabilistic sensing may fail to sense certain events occurring around the devices.



(a) Sensing Models (b) Network Lifetime (in minutes)

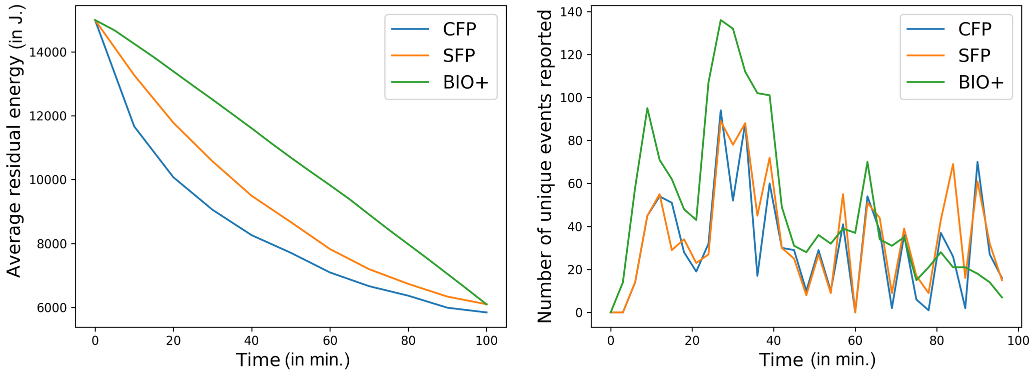
Figure 6.9. *bioSmartSense*+ Vs. *bioSmartSense*.

Figure 6.9(a) shows the comparison of the number of unique events sensed by the system using the Elfes and boolean sensing models on the Waze data. Clearly, the curves for boolean and Elfes are almost overlapping, implying that very few events are lost as a result of probabilistic sensing. This is because a single event is typically sensed by several IoT devices. As a consequence, if a device fails to sense a certain event due to probabilistic sensing, another device is likely to sense it. We infer that regardless of boolean or Elfes sensing, *bioSmartSense+* is capable of sensing events with similar effectiveness.

6.4.6. Effect of Event Priority on Network Lifetime. In this experiment we study how the longevity of the IoTNet under *bioSmartSense+* (i.e., IG function-based prioritized event reporting). We consider the overall network energy budget be $\mathcal{B} = 3 \times 10^5$ J. Recall that the simulation is revoked once the total energy expended for sensing and reporting exceed $\mathcal{B}$. We show the effect of IG based event prioritization on the network lifetime (defined in Section 6.2.3). Figure 6.9(b) shows that for both real and simulated data, the IG function improves the overall network lifetime.

6.4.7. Comparison with the State-of-the-Art. We compare *bioSmartSense+* with the distributed data collection proposed by Capponi *et al.* [64]. Unlike *bioSmartSense+*, their distributed framework considers sensors embedded in hand-held devices. We compare *bioSmartSense+* with two distinct data collection policies [64] using the events generated from Waze data: (i) *Collector Friendly Policy (CFP)* which maximizes data collection utility, and (ii) *Smartphone Friendly Policy (SFP)* which emphasizes on energy efficiency of smart devices over event reporting.

Figure 6.10(a) shows that the average residual energy of *bioSmartSense+ (BIO+)* is significantly larger than *CFP* and *SFP*. Clearly, the inclusion of the IG function-based event prioritization leads to an improvement in energy efficiency. Moreover, note that the curves corresponding to *CFP* and *SFP* are not linear due to non-uniform event distribution in the real data set. Consequently, different IoT devices report varying number of reports to the base station, incurring varying communication overhead. The plot corresponding to *bioSmartSense+* is near-linear because only a small fraction of devices are chosen to report sensed events to the base station and their individual variations in residual energy has little effect on overall residual energy.



(a) Average Residual Energy (b) Number of Unique Events Reported

Figure 6.10. Comparison with state of the art.

In Figure 6.10(b), we show that *bioSmartSense*+ exhibits the highest event reporting rate through most of the simulation duration. However, the residual energy decreases with time, our framework attempts to conserve energy by invoking the IG function for selective reporting of high priority events. Consequently, we observe a noticeable drop in the number of reported events.

6.5. INFERENCES

In this section we propose a probabilistic data collection framework for priority-based event reporting in IoT-based environments, called *bioSmartSense*+. The proposed framework is capable of saving energy of IoT devices with limited residual energy, by allowing them to preferentially report high priority events. We compare the effects of a boolean sensing and a more realistic probabilistic (Elfes) sensing model on the performance of *bioSmartSense*+. Furthermore, our rigorous experimental study on real as well as synthetic data demonstrate that when compared to state-of-the-art data collection frameworks, *bioSmartSense*+ exhibits high priority-based event reporting at a considerably low event reporting energy cost, thereby maximizing network lifetime. In the future, we shall extend this work to incorporate mobility of IoT devices.

7. BIO-INSPIRED DISASTER RESPONSE NETWORK

In this section, we conceive a novel energy-efficient yet robust DRN topology, which is inspired from biological networks, namely *transcriptional regulatory networks* (TRNs). Our work is motivated by the following structural similarities between the TRN and DRN topologies: First, a TRN possesses few well-connected entities (usually proteins), called *Transcription Factors (TFs)*, that control bulk of the protein interactions within the network [133]. Analogously, a DRN also possesses few well-connected entities, such as CC and PoIs, which are central to its information flow. Second, a large fraction of TRN nodes consist of regulated genes that are loosely connected to the TFs. Such regulated genes are similar to the survivors in a DRN that make up a large part of the network and communicate with few PoIs or CC.

In this paper we propose to utilize TRN as model for designing energy-efficient yet robust DRN topology, termed *Bio-DRN*, which mimics the graph robustness of TRNs. Specifically, the Bio-DRN is a subgraph of originally formed DRN (in short, Orig-DRN) topology, constructed by one-to-one mapping between structurally similar genes and DRN components (viz., survivors, volunteers, PoIs and CC). Let us briefly discuss the key contributions of this work.

- We design, for the first time, an energy-efficient and robust DRN topology, termed *Bio-DRN*, which is inspired from a biological network, i.e., TRN.
- We formulate the Bio-DRN topology construction (BioTopoC) as an integer linear programming (ILP) optimization problem, and show that it is NP-hard.
- We propose a sub-optimal heuristic that intelligently constructs Bio-DRN through one-to-one mapping between structurally similar genes and DRN components.

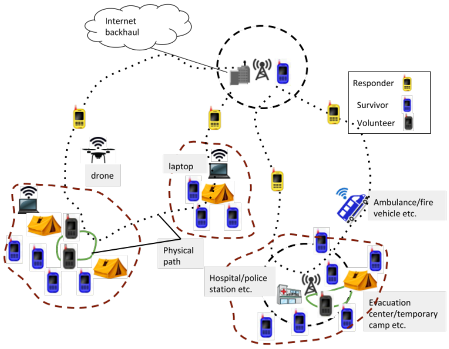

Internet backhaul
Responder
Survivor
Volunteer
drone
laptop
Ambulance/fire vehicle etc.
Physical path
Hospital/police station etc.
Evacuation center/temporary camp etc.

Figure 7.1. System model. A representative post-disaster scenario. Blue, yellow, and black colored smart devices respectively denote survivors, responders, and volunteers.

- Through extensive simulation study on a real disaster prone region in Bhaktapur, Nepal, we demonstrate that the Bio-DRN topology notably outperforms several other approaches (See Section 7.3 for details) in terms of both energy efficiency and network robustness, while guaranteeing the desired quality of service (QoS) requirements, i.e., packet delivery ratio and network latency.

7.1. NETWORK MODEL AND ASSUMPTIONS

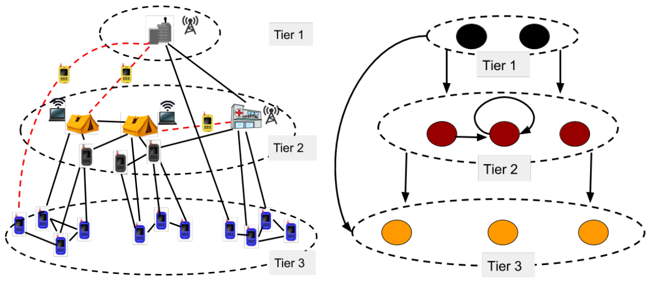
As shown in Figure 7.1, we consider a large-scale post disaster scenario, such as an earthquake. We first discuss the key components, followed by the network model.

7.1.1. Key Components. Let us discuss the key players in the post-disaster scenario.

7.1.1.1. Survivors. The affected individuals equipped with wireless devices, such as smart phones, which are capable of short range ad-hoc communication (via bluetooth). We assume that each survivor has a disaster application installed on his device (such as Surakhshit [147]) that allows him to (a) establish ad-hoc communication, and (b) exchange situational or rescue/relief related information in forms of text, image, audio and video. Such survivors, either static or mobile, usually remain confined within their respective PoI due to unsafe outside environment [51]. There exists a certain subset of survivors, termed *volunteers*, who usually move (shown as higlighted green lines in Figure 7.1) within the vicinity of its own PoI and provide relief and services to other survivors [50].

7.1.1.2. Points of Interest (PoIs). Certain safe geographical places, such as shelter points, schools, parks, hospitals, preexisting evacuation centers, temporary camps etc., where the survivors gather in the aftermath of a disaster. We assume that these PoIs possess communication equipments, e.g., a laptop or unimpaired WiFi router/tower. Each PoI location is fixed.

7.1.1.3. Coordination Center (CC). A controlling station that coordinates the entire rescue/relief operations in the disaster area. The CC is equipped with WiFi router/tower. All the data generated by the survivors are eventually delivered to the unique CC. The location of CC is also fixed. For ease of presentation we consider a unique CC, however, our proposed system model can easily incorporate multiple CCs.



(a)

(b)

Figure 7.2. Three tier topology in TRN and DRN. (a) Orig-DRN: Components and three tier communication structure. A black solid line denotes a communication link e^t (over a common ad-hoc or WiFi medium) in the Orig-DRN at a time slot t. A dotted red line denotes an indirect link via a responder., and (b) Three tier topology of TRN.

7.1.1.4. Responders. Members of rescue and relief operation groups, medical teams, police and fire vehicles, which patrol one or more PoIs over physical paths (shown as dotted black lines in Figure 7.1). We consider that each responder is equipped with a wireless device (with ad-hoc mode) or a WiFi router.

The difference between a responder and a volunteer is that a responder travels back and forth between the PoIs and the CC, whereas a volunteer moves within the vicinity of its own PoI. Hereafter, we refer to a survivor, volunteer, PoI, or CC (equipped with a wireless device or a WiFi router), as a *node*. Note that responders are not considered as nodes, rather they act as indirect communication links (or data mules) between two nodes in the network.

7.1.2. Network Model. Due to the mobility of nodes, intermittent connectivity and failure of communication equipments (due to hardware faults or energy exhaustion), the Orig-DRN can be modeled as a time-evolving graph. Consider the total time duration T, say 12 hours, to be divided into discrete time slots. Then, at a given time slot $t \in H$, the Orig-DRN topology is a directed graph $G_d^t = (V_d^t, E_d^t \cup \mathcal{E})$. V_d^t is the set of nodes comprising survivors, PoIs, volunteers, and CC; E_d^t is the set of communication links, where each $e^t(u, v) \in E_d^t$ indicates that nodes u and v have come within the transmission range (over a common ad-hoc or WiFi medium) for at least a prespecified duration of time within the current time slot t; and $\mathcal{E}$ is the set of indirect links due to communication between a pair of nodes via a responder. Hereafter we drop t from all notations, because the proposed Bio-DRN topology is constructed from Orig-DRN (G_d^t) independently at each time slot t (See Section 7.2) . Based on the functional roles of DRN components, the node set V_d can be classified into three distinct tiers (as illustrated in Figure 7.2(a)): **tier 1** contains the unique CC; **tier 2** comprises the set of PoIs and volunteers; and **tier 3** contains the set of survivors.

7.2. BIO-DRN TOPOLOGY

Here we formulate the Bio-DRN topology construction (BioTopoC) problem, and show that it is NP-Hard. Then, we present a novel (sub-optimal) heuristic for the same.

7.2.1. Problem Formulation. We formulate the BioTopoC problem as an Integer Linear Programming (ILP) optimization problem. BioTopoC aims at constructing a Bio-DRN topology as a common subgraph of the Orig-DRN and input TRN (e.g., Yeast TRN) topologies, while preserving the graph properties of TRN, particularly *low graph density* and *motif abundance*. These two graph properties of TRN are of particular interest in our context because of the following reasons: (a) low graph density ensures fewer communication links, which translates into fewer message replications and forwarding, and thus improved *energy efficiency*, and (ii) high motif abundance renders alternative communication pathways, which improve the topological *robustness* against node failures. In other words, the key objective of BioTopoC is to construct a common subgraph (i.e., Bio-DRN topology), which maximizes the *FFL motif count* and *number of common edges* between the Orig-DRN and TRN topologies, in order to enhance the network robustness and energy efficiency.

To formulate the BioTopoC problem as an ILP optimization, we define a binary variable y_{ik} that represents the binary decision to map a node $v_i \in V_g$ to a unique node $v_k \in V_d$ (denoted by $v_i \leftrightarrow v_k$).

We introduce a second binary variable x_{ij} which equals 1, if there is an edge $e_{ij} \in E_g$, and there exists an edge $e_{kl} \in E_d$, such that $v_i \leftrightarrow v_k$ and $v_j \leftrightarrow v_l$, where $v_i, v_j \in V_g$ and $v_k, v_l \in V_d$.

Finally, we define a third binary variable z_{ijp} which equals 1, if there are three edges $e_{ij}, e_{jp}, e_{pi} \in E_g$ and there exists three corresponding edges $e_{kl}, e_{lq}, e_{qk} \in E_d$, such that $v_i \leftrightarrow v_k$, $v_j \leftrightarrow v_l$ and $v_p \leftrightarrow v_q$ where $v_i, v_j, v_p \in V_g$ and $v_k, v_l, v_q \in V_d$. In other words, z_{ijp} determines whether a motif exists in V_g if there exists a motif in V_d, such that each participating node of the motif in V_g is uniquely mapped to the corresponding node of the motif in V_d.

Expressions 7.1-7.6 show the ILP formulation. The objective function of the BioTopoC problem has two parts. The first part maximizes the cardinality of common edges and the second part maximizes the number of motifs between two graphs G_g and G_d. The variables $0 \leq \alpha, \beta \leq 1$ are control parameters. Eq. 7.2 constrains the Bio-DRN topology to have the exact number of nodes as that of Orig-DRN graph.

Constraint 7.3(a) shows that every node in the TRN graph is mapped to at most one node in the DRN graph. Similarly, constraint 7.3(b) depicts that for every node in Orig-DRN graph, there is at most one node in TRN graph mapped to it.

Given $y_{ik} = 1$, inequality 7.4 enforces two conditions: (a) if $x_{ij} = 0$, there exists no neighbor $v_l \in N(k)$ (the neighbor list of node v_k) such that $v_j \leftrightarrow v_l$, and (b) if $x_{ij} = 1$, then, there must exist a neighbor $v_l \in N(k)$ such that $v_j \leftrightarrow v_l$.

Inequalities in 7.5 show that for any node $v_i \in V_g$, the binary variable (for motif count) z_{ijp} is 1 if e_{ij}, e_{jp}, and $e_{pi} \in E_g$ between the mapped nodes v_i, v_j and v_p, are also mapped.

Finally, expression in 7.6 represent the decision variables assuming values 0 or 1.

$$\textbf{Maximize}\ \alpha \sum_{e_{ij} \in E_g} x_{ij} + \beta \sum_{v_i \in V_g} \sum_{v_j \in V_g} \sum_{v_p \in V_g} z_{ijp} \tag{7.1}$$

$$\sum_{v_k \in V_d} y_{ik} = |V_d|, \ \ \exists v_i \in V_g \tag{7.2}$$

$$(a) \sum_{v_k \in V_d} y_{ik} \leq 1, \ \forall v_i \in V_g, \ (b) \sum_{v_i \in V_g} y_{ik} \leq 1, \forall v_k \in V_d \tag{7.3}$$

$$x_{ij} + y_{ik} \leq 1 + \sum_{l \in N(k)} y_{jl}, \ \ e_{ij} \in E_g, \ v_k \in V_d \tag{7.4}$$

$$z_{ijp} \leq x_{ij}, \ \ z_{ijp} \leq x_{jp}, \ \ z_{ijp} \leq x_{pi} \tag{7.5}$$

$$y_{ik}, x_{ij}, z_{ijp}, \in \{0, 1\}\, \forall v_i \in V_d, v_k \in V_g, \forall e_{ij}, e_{jp}, e_{pi} \in E_g \tag{7.6}$$

The BioTopoC problem is NP-hard. We provide a reduction from the *Maximum Common Edge Subgraph* (MCES) problem [148]. Let us consider a generic instance of MCES. Given two graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$, the goal of the MCES problem is to determine a subgraph $G_{12}(V_{12}, E_{12})$ of maximum number of edges, which is isomorphic to a common subgraph of both G_1 and G_2.

We reduce this MCES problem to an instance of BioTopoC problem as follows. Consider an Orig-DRN graph $G_d(V_d, E_d)$ with $V_d = V_1$ and $E_d = E_1$ and an input TRN graph $G_g(V_g, E_g)$ with $V_g = V_2$ and $E_g = E_2$. Now, let us assign $\alpha = 1$ and $\beta = 0$ in the objective function for the BioTopoC problem (see Expression 7.1). For this instance, BioTopoC constructs a Bio-DRN graph $G_d^{bio}(V_d^{bio}, E_d^{bio})$ that is the subgraph with maximum number of edges common to graphs G_d and G_g. In order to prove that G_d^{bio} is also the optimal solution for the MCES problem, we need to show that the additional constraint $|V_d^{bio}| = |V_d|$ does not limit the solution space. However, if a node v is in V_d^{bio}, but not in V_{12}, it means that v is an isolated node, otherwise it would have been included in V_d^{bio} to further maximize the number of common edges. As a result, by removing isolated nodes, the solution of BioTopoC can be translated to the solution of the MCES problem. Therefore, if we are able to solve BioTopoC in polynomial time, we are also able to solve MCES in polynomial time. Since MCES is NP-Complete, BioTopoC is NP-hard.

7.2.2. Proposed Heuristic. In this subsection, we present a polynomial-time heuristic that constructs a Bio-DRN topology, given the Orig-DRN topology and input TRN (termed, Orig-TRN) topology.

In every timeslot $t \in H$, the proposed heuristic operates in three steps (Figure 7.3). First, it generates reference TRN (Ref-TRN) G_g^{ref}, a subgraph of the Orig-TRN G_g which acts as a template for construction of the Bio-DRN. Then, the algorithm computes Blondelś similarity metric [149] to determine the neighbor-based similarity between each node pair in G_g^{ref} and G_d. Finally, it utilizes the Hungarian Algorithm [150] to construct the Bio-DRN G_d^{bio}, by uniquely mapping structurally similar nodes in G_g^{ref} and G_d, such that the overall node pair similarity score is maximized. The details of the algorithm are presented below.

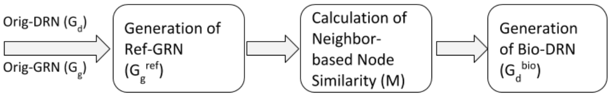

Orig-DRN (G_d)
Orig-GRN (G_g)
Generation of Ref-GRN (G_g^{ref})
Calculation of Neighbor-based Node Similarity (M)
Generation of Bio-DRN (G_d^{bio})

Figure 7.3. Overview of the Proposed Heuristic.

7.2.2.1. Generation of Ref-TRN (G_g^{ref}). Algorithm 4 generates a Ref-TRN, $G_g^{ref} = (V_g^{ref}, E_g^{ref})$, where $V_g^{ref} \subseteq V_g$, $E_g^{ref} \subseteq E_g$, and $|V_g^{ref}| = |V_d|$. An empty G_g^{ref} is initialized in Line 4. In Lines 5-6, for both TRN and DRN graphs, nodes in i^{th} tier (where $i = 1, 2, 3$)

are sorted in non-increasing order of node motif centrality Δ. In Lines 8-14, the algorithm adds any node $v \in V_g^2$ from G_g to G_g^{ref}, if there exists an unvisited node $u \in V_d^2$ such that $|V_g^1(v)| \geq |V_d^1(u)|$ and $|V_g^3(v)| \geq |V_d^3(u)|$, where $V_g^1(v)$ and $V_d^1(u)$ (and $V_g^3(v)$ and $V_d^3(u)$) denote the set of neighboring nodes of node v and u in tier 1 (and tier 3) of TRN and Orig-DRN graphs. The node set $V_g^1(v)$ and $V_g^3(v)$ are also added to V_g^{ref}. This step ensures that G_g^{ref} contains the most motif central nodes of sufficiently high degree, facilitating the subsequent mapping process (Step 3). In Lines 16-19, G_g^{ref} is generated by adding each edge $e(u, v)$ to G_g^{ref} if $e(u, v) \in E_g$ and nodes u, v belong to the mapped nodeset V_g^{ref}.

Algorithm 4 Generation of Ref-TRN

```
 1: Input: $G_g, G_d, \Delta$
 2: Output: $G_g^{ref}$
 3: procedure GEN-REF-TRN($G_g$, $G_d$)
 4:     Initialize $G_g^{ref}$ where $V_g^{ref} = \phi$ and $E_g^{ref} = \phi$.
 5:     for $i = 1$ to 3 do
 6:         Sort $i^{th}$ tier DRN and TRN node sets, i.e., $n_d^i$ and $n_g^i$ in decreasing order of motif centrality $\Delta$, respectively
 7:     //Add nodes to reference TRN
 8:     for DRN node $u \in V_d^2$ do
 9:         for TRN node $v \in V_g^2$ do
10:             Compute the set of neighbor nodes of $u$ in tiers 1 and 3 denoted by $V_d^1(u)$ and $V_d^3(u)$, respectively.
11:             Compute the set of neighbor nodes of $v$ in tiers 1 and 3 denoted by $V_g^1(v)$ and $V_g^3(v)$, respectively.
12:             if DRN node $u$ not visited and $|V_g^1(v)| \geq |V_d^1(u)|$ and $|V_g^3(v)| \geq |V_d^3(u)|$ then
13:                 $V_g^{ref} = V_g^{ref} \cup v \cup V_g^1(v) \cup V_g^3(v)$
14:                 Mark node $u$ as visited.
15:     //Add edges to reference TRN
16:     for TRN node $u \in V_g^{ref}$ do
17:         for TRN node $v \in V_g^{ref}$ do
18:             if $u \neq v$ and edge $e(u, v) \in E_g$ then
19:                 $E_g^{ref} = E_g^{ref} \cup e(u, v)$
20:     Return $G_g^{ref}$
```

7.2.2.2. Calculation of node similarity. *Blondel's node similarity* [149] is a metric to measure the similarity between two nodes u and v, each belonging to different input graphs, based on the similarity scores of their respective neighbors. In our context, we utilize the Blondel's node similarity to perform one-to-one mapping between similar nodes of G_g^{ref} and G_d belonging to the same tiers. Algorithm 5 calculates the similarity matrix M of dimension $|V_d| \times |V_g^{ref}|$. Each entry $M_{u,v}$ denotes the similarity value (on the scale of 0-1) between node $u \in V_d$ and node $v \in V_g^{ref}$. In Lines 7-14, M is iteratively calculated (using Eq. 7.7), until any of the two conditions are met: (i) the improvement in similarity score between successive iterations i and $(i + 1)$, is less than a predetermined threshold (ϵ), i.e., $M^{i+1} - M^{i} \leq \epsilon$, or (ii) the prespecified maximum number of iterations $maxIter$ (= 100 in our experiments) is reached.

Algorithm 5 Calculation of Node Similarity

```
 1: Input: $G_g^{ref}, G_d, maxIter, \epsilon$
 2: Output: $M$
 3: procedure GET-NODE-SIM($G_g^{ref}, G_d, maxIter, \epsilon$)
 4:     Initialize iteration count, $i = 0$
 5:     Define similarity matrix $M_{|V_d| \times |V_g^{ref}|}$
 6:     Initialize $M^i_{u,v} = 0.1, \quad \forall u \in V_d, v \in V_g^{ref}$
 7:     while $i \leq maxIter$ do
 8:         for $u \in V_d$ do
 9:             for $v \in V_g^{ref}$ do
10:                 Compute $\tilde{M}^{i+1}_{u,v}$ using Eq. 7.7
11:                 Calculate $M^{i+1}_{u,v} = \frac{\tilde{M}^{i+1}_{u,v}}{\sum_{u,v} \tilde{M}^2_{u,v}}$
12:         if $M^{i+1} - M^{i} < \epsilon$ then
13:             break
14:         $i = i + 1$
15:     Return M
```

In Line 11, the updated neighbor-based node similarity value M_{uv} is computed by the normalization of $\tilde{M}_{uv}$ (obtained from Eq. 7.7) by $\sum_{u,v} \tilde{M}^2_{u,v}$.

$$\tilde{M}^{i+1}_{u,v} = \sum_{\substack{r:(r,u) \in E_d \\ s:(s,v) \in E_g^{ref}}} M^i_{r,s} + \sum_{\substack{r:(u,r) \in E_d \\ s:(v,s) \in E_g^{ref}}} M^i_{r,s} \tag{7.7}$$

7.2.2.3. Generation of Bio-DRN. After calculating node similarity, our heuristic determines one-to-one mapping between each node $u \in V_d^i$ and a node $v \in V_g^{ref,i}$, such that both u, v belong to the same i^{th} tier of their respective graphs and the sum of pairwise similarity scores between the mapped nodes is maximized. This mapping problem can be modeled as a well-known graph problem called *minimum weighted bipartite matching* and can be optimally solved using the Hungarian algorithm [150], wherein the input is the additive inverse of the Blondel's similarity matrix M. The Hungarian algorithm has a time complexity of $O(|V|^3)$, where $|V|$ is the minimum number of nodes of both the input graphs.

Now let us briefly describe the generation of Bio-DRN topology (refer Algorithm 6). In Line 4, Bio-DRN graph G_d^{bio} is initialized with V_d nodes and empty edge set. In Line 6, $\mathbf{M}^i$ is defined as a sub-matrix of M for i^{th} tier nodes. Each element in $\mathbf{M}^i$ (denoted by $\mathbf{M}^i_{u,v}$)

is an additive inverse of the similarity score between nodes u and v $M^i_{u,v}$, i.e., $\mathbf{M}^i_{u,v} = -M^i_{u,v}$, where $u \in V^i_d$ and $v \in V^{ref,i}_g$. In Line 7, for each tier i, the Hungarian algorithm is invoked to calculate the mapping function $f : V^i_d \rightarrow V^{ref,i}_g$. Finally, in Lines 9-12, each edge $e(u, v)$ is added to E^{bio}_d, if (i) there exists $e(u, v) \in E_d$ and (ii) a path between mapped nodes $f(u)$ and $f(v)$ exists in E^{ref}_g, i.e., $has\_path(f(u), f(v)) \in E^{ref}_g$. The idea behind this step is to preserve the structural properties of TRN (i.e. G^{ref}_g) by embedding an edge in G_d into a path in G^{ref}_g, similar to the approach discussed in [46].

Algorithm 6 Generation of Bio-DRN

```
 1: Input: $G_d, G^{ref}_g, M$
 2: Output: Bio-DRN $G^{bio}_d$
 3: procedure GEN-BIO-DRN($G_d, G^{ref}_g, M$)
 4:     $V^{bio}_d = V_d, E^{bio}_d = \phi,$
 5:     for tier i = 1 to 3 do
 6:         $\mathbf{M}^i = \{\mathbf{M}^i_{u,v} | \mathbf{M}^i_{u,v} \subset M, u \in V^i_d, v \in V^{ref,i}_g\}$
 7:         f = HUNG-ALG($\mathbf{M}^i$) //Mapping Function
 8:     //Add edges to Bio-DRN graph
 9:     for $u \in V_d$ do
10:         for $v \in V_d$ do
11:             if $e(u, v) \in E_d$ and $has\_path(f(u), f(v)) \in E^{ref}_g$ then
12:                 $E^{bio}_d = E^{bio}_d \cup e(u, v)$
```

Time Complexity: The input to the mapping heuristic includes the motif centrality Δ of input TRN and Orig-DRN nodes, which incurs $O(|V_g|^3)$ and $O(|V_d|^3)$, respectively. Algorithm 4 generates G^{ref}_g and incurs the time complexity of $O(|V_d| \times |V_g|)$. The time complexity for Algorithm 5 is given by $O(maxIter \times |V_d|^4)$ since each computation of Eq. 7.7 requires $O(|E_d|)$ $(= O(|V_d|^2)$ time, and this is repeated $|V_d|^2$ times. Algorithm 6 executes the Hungarian algorithm which has a time complexity of $(|V_d|^3)$. We exclude the computation of motif centrality of Orig-DRN and TRN nodes from the total time complexity, since this step is executed only once. As the Ref-TRN and Orig-DRN have the same number of nodes (i.e., $|V^{ref}_g| = |V_d|$), the total time complexity of the mapping algorithm is $O(|V_g| \times |V_d| + (maxIter \times |V_d|^4) + |V_d|^3)$.

Discussion on the heuristic. Depending on the number of mapped nodes, the proposed heuristic may not always generate the Bio-DRN topology with exactly the same number of nodes as the Orig-DRN. Bio-DRN may have fewer nodes in the following two scenarios:

- *Unmapped nodes:* For each tier $i = 1, 2, 3$, the Hungarian algorithm maps $min(|V_d^i|, |V_g^{ref,i}|)$. If $|V_g^{ref,i}| < |V_d^i|$, the number of unmapped Orig-DRN nodes in the i^{th} tier is $|V_d^i| - min(|V_d^i|, |V_g^{ref,i}|)$.

- *Isolated nodes:* As shown in Line 11 of Algorithm 6, for each pair of nodes u and v in the Bio-DRN, an edge $e(u, v) \in E_d$ is added to Bio-DRN if there exists an path between the corresponding mapped nodes in the Ref-TRN, i.e., $e(f(u), f(v)) \in E_g^{ref}$. Given any $u \in V_d$, if there is no edge $e(f(u), f(v)) \in E_g^{ref}$ for all $v \in V_d$, node u remains isolated.

We address these issues in the following ways. First, all the isolated and unmapped nodes from Orig-DRN topology are added to Bio-DRN. Second, at most κ shortest paths existing between each unmapped (or isolated) node and the CC in Orig-DRN is retained in Bio-DRN. We take the value of $\kappa = 2$, for it preserves the low graph density of Bio-DRN topology, while ensuring two communication pathways (or robustness) between unmapped/isolated node and CC. At the end of these steps, Bio-DRN is a connected graph with the same number of nodes as Orig-DRN provided that the Orig-DRN is connected.

Inferences of Orig-DRN Topology. From our discussion so far, we infer that the proposed heuristic for the construction of Bio-DRN topology requires two topologies: TRN and Orig-DRN. While TRN is provided as an input, Orig-DRN topology is inferred in the following manner:

We consider that the CC, which is potentially the most well-connected entity in the Orig-DRN, observes the Orig-DRN topology over time. Note that the collected Orig-DRN topology information may not always be the most recent information (i.e., corresponding to current time slot). However, our approach for the realization of Bio-DRN topology still remains viable and practical, thanks to the relatively steady nature of Orig-DRN over time. Recall that Orig-DRN, though a time-evolving graph, does not change drastically from one time slot to the next. This is mainly because the mobility of survivors, which constitutes the majority of Orig-DRN, are restricted to a certain PoI; additionally, locations of PoIs and CC are fixed and the responders are prespecified to patrol certain subset of PoIs. Refer to Section 7.1 for details.

7.3. PERFORMANCE EVALUATION

7.3.1. Simulation Setting. We simulate Bio-DRN in the Opportunistic NEtwork (ONE) simulator [151] on top of the post-disaster mobility model (PDM) as outlined in Section 7.3.1.1. For our experiments, we consider a real disaster prone region Bhaktapur, Nepal, over an area of 5×5 sq. km (see Figure 7.4). The map of Bhaktapur is extracted from OpenStreetMap using Overpass API [152] and then converted to the Well-Known Text (.wkt) format using osm2wkt [153]. The .wkt file is then utilized by the PDM model in the ONE Simulator.

7.3.1.1. Post-disaster mobility model. We utilize the post-disaster mobility model originally proposed by Uddin et al. [50] to evaluate the Bio-DRN topology in a post-disaster setting. The CC and PoIs are fixed for the entire simulation time period. Conversely, each survivor is randomly assigned to a unique PoI and moves within the PoI's boundary with a speed randomly chosen in the interval [0.5 - 1.5] meters/sec, representative of typical walking pace. Survivors wait a random time of [2 - 5] min. between successive movements. Volunteers and responders follow the shortest path map based mobility model [154] between PoIs and the CC. The speed is chosen in [2 - 10] meters/sec, while the waiting time at each visited location is [2 - 10] min.

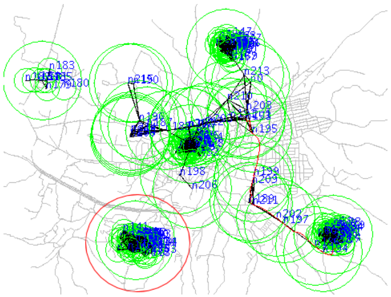
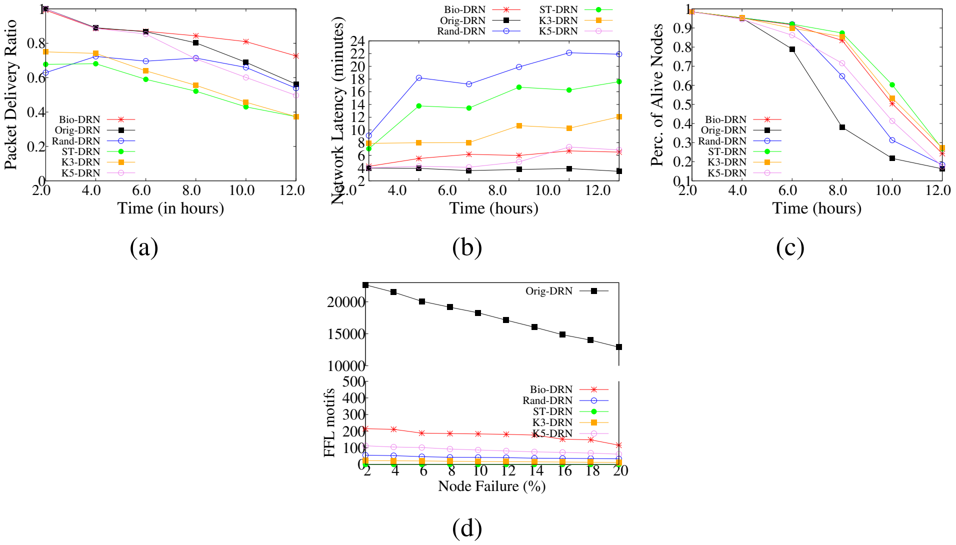


Figure 7.4. Post disaster scenario. A snapshot of the post-disaster scenario. A red circle denotes the boundary of a PoI and the red line denotes a certain prespecified route of a responder. The larger green circle is the communication range for WiFi routers (500 meters) and the smaller one is for the ad-hoc (50 meters).

7.3.1.2. Energy consumption model. We now discuss the energy consumption model adopted in the simulation experiments. The energy model estimates the total energy expenditure for any node i, as described in the following.



(a)

(b)

(c)

(d)

Figure 7.5. Bio-DRN performance 1. (a) PDR vs Time, (b) Network latency vs Time, and (c) Perc. of alive nodes vs Time, and (d) Motif count vs Perc. of node failure.

7.3.1.2.1. Message transmission. The energy consumed by transmitting a message m of size L, from a node i to a neighboring node j, is given by $E^{tx}(i,m) = e^{tx} \times t_{ij}$, where e^{tx} is the transmit energy per unit time, and t_{ij} is the message transmission time, defined as the time taken to deliver the entire message m from node i to j.

7.3.1.2.2. Message reception. The energy consumed by i for receiving a message m of size L, from a neighboring node j is given as $E^{rx}(i) = e^{rx} \times t_{ji}$, where e^{rx} is the reception energy per unit time, and t_{ji} is the message transmission time.

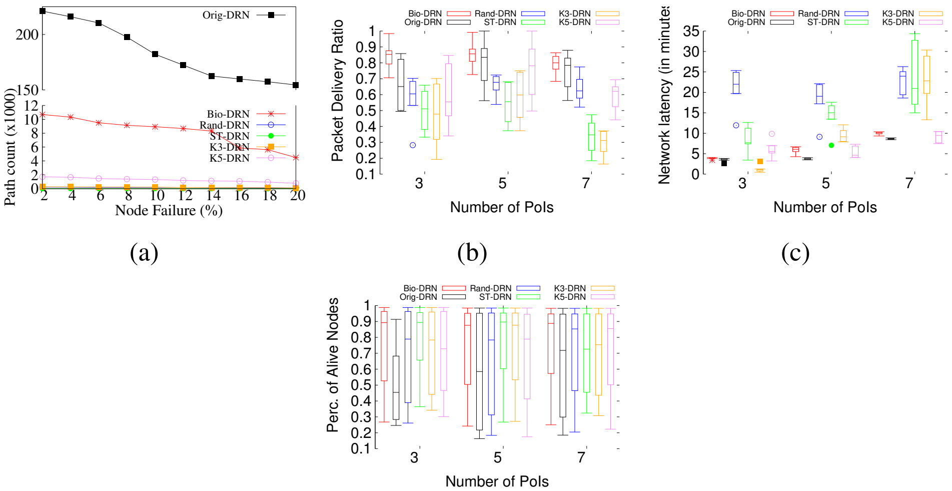
7.3.1.2.3. Scanning for neighboring devices. The energy consumed by periodically scanning for neighboring nodes is calculated as $E^{sc}(i) = e^{sc} \times t^{sc}$, where e^{sc} is the energy consumed per unit time and t^{sc} is the predetermined scan time interval. Note that e^{sc} may be up to 5 times higher than the transmission energy e^{tx} [155].

7.3.1.2.4. Idle state. The energy consumed when the node is idle, is given by $E^{id} = e^{id} \times t^{id}$, where e^{id} is the idle energy per unit time and t^{id} is the idle time period.

For our experiments, we consider the following realistic energy values: $e^{tx} = e^{rx} =$ 0.6 J (transmission/reception), $e^{sc} = 3$ J (scanning), and $e^{id} = 0.005$ J (idle), where J stands for Joules. The scan interval e^{sc} is pre-specified and taken as 60 seconds. Finally, each survivor has an initial energy in the interval [0.8 - 1.2] kJ, while a volunteer, PoI or CC has a high initial energy in the interval [3 - 5] kJ.

7.3.1.3. Routing protocol. We utilize the Epidemic routing [49], which is a standard flooding-based routing protocol in DRNs (and DTNs). According to this protocol, a certain node replicates and forwards messages to every encountering node. As previously mentioned, in this context the energy expenditure at a node is quantifiable by the number of links it shares.

All the experiments, unless otherwise stated, are performed with one CC, 5 PoIs, each with [30 - 50] survivors, [1 - 5] volunteers, and 30 responders. Each survivor generates data traffic with a prespecified message generation rate of 1 packet per [1 – 2] minutes. The values of other important parameters are: (i) message packet size: [250 – 500] Kb (ii) time-to-live: 1 hour, and (iii) data rate, range of ad-hoc and Wi-Fi medium: (2 Mbps, 50 meter), and (8 Mbps, 500 meter), respectively. The simulation time period $H = 12$ hr, and duration of each time slot $t \in H$ is 15 minutes (See Section 7.1).



(a) (b) (c)

(d)

Figure 7.6. Bio-DRN performance 2. (a) Path count vs Perc. of node failure, (c) PDR vs Num. of PoIs, (c) Network latency vs Num. of PoIs, and (d) Perc. of alive nodes vs Num. of PoIs. A certain boxplot bar depicts the range of (PDR, perc. of alive nodes or latency) values over time, through their quartiles. A box plot may have line extending vertically from the box indicating variability outside the upper and lower quartiles. Outliers are plotted as individual points.

7.3.2. Simulation Experiments.

Besides Orig-DRN, we compare the energy efficiency, QoS and network robustness of the Bio-DRN topology against the following three standard network topologies:

- *ST-DRN*- spanning tree, constructed by removing surplus edges from Orig-DRN.
- *Rand-DRN*- subgraph with same graph density as Bio-DRN, created by random edge removal from Orig-DRN.
- *K-DRN*- subgraph of Orig-DRN, constructed by pruning edges, if and only if both their end nodes have more than K neighbors. In our experiments, we consider $K = 3, 5$. K3-DRN has comparable graph density as Bio-DRN, while K5-DRN is almost twice as dense (See Table 7.1).

Table 7.1. Edge count for 5 PoIs.

	Orig-DRN	Bio-DRN	Rand-DRN	ST-DRN	K3-DRN	K5-DRN
No. of edges	2083	352	352	181	297	638
Avg. Diameter	5.5	6	12.0	24.0	14.0	9.0

The rationale behind considering ST-DRN is that it is the sparsest DRN topology and thereby promises the highest energy efficiency. In contrast, K-DRN with higher *K* will be likely to offer higher connectivity in the event of node failures, at the expense of poor energy efficiency.

7.3.2.1. Energy efficiency and QoS analysis. In the following subsections, we evaluate Bio-DRN against all the aforementioned topologies, in terms of following three performance metrics: (a) *Packet delivery ratio* - the fraction of total unique messages successfully delivered at the CC to the total generated messages at the survivor nodes or vice versa, (b) *Network latency* - the average delay incurred in delivering the messages from the survivors to the CC. Recall that QoS is measured in terms of PDR and network latency., and (c) *Energy efficient* - percentage of alive nodes. *We simulate the failure of* 2% *nodes (except CC) after every one hour (summing up to* 20% *in total) to imitate random failures due to hardware faults or environmental adversities.*

7.3.2.1.1. Packet Delivery Ratio (PDR). Figure 7.5(a) shows that PDR achieved by Bio-DRN is notably better than other topologies. This is because Bio-DRN offers multiple paths during node failures (by preserving FFL motifs as shown in Figure 7.5(d)) between any survivor-CC pair. Note over time (every 1 hour), 2% randomly chosen nodes are failed in addition to the dying nodes due to energy depletion (See Figure 7.5(c)). In addition, we observe that Bio-DRN sustains its steady PDR by ensuring that a large fraction of nodes are alive over time (See Figure 7.5(c)). Among other topologies, ST-DRN and K3-DRN, the two sparsest topologies, yield poor PDR, as a result of irregular network partitions due to node failures. Finally, we attribute the gradual decline in PDR of Orig-DRN, K5-DRN, and Rand-DRN to the rapidly dying nodes due to energy depletion (See Figure 7.5(c)).

7.3.2.1.2. Network Latency. Figure 7.5(b) shows that Bio-DRN exhibits better network latency than that of topologies with similar (or lower) graph densities, i.e., ST-DRN, K3-DRN, Rand-DRN. This is because Bio-DRN, though sparse, preserves the small-world property (i.e. low diameter) of TRN, which ensures short communication pathways between survivors and CC (See Table 7.1). By the same token, ST-DRN, K3-DRN and Rand-DRN suffer from very high network latency. Though Orig-DRN and K5-DRN offers comparable or better latency, they suffer from poor energy efficiency (See Figure 7.5(c)).

7.3.2.1.3. Energy efficiency. Figure 7.5(c) shows that Bio-DRN exhibits notable improvement in terms of energy efficiency compared to all standard topologies, except ST-DRN and K3-DRN. The reasons are the following: Bio-DRN possesses (i) fewer communication links, which translate into lower energy expenditure at a certain node, and moreover, (ii) lower network *diameter* implying that only a few intermediate nodes consume energy to transmit packets between a certain survivor-CC node pair (See Table 7.1). Recall diameter of a graph is the length of the longest shortest path between any two node in the considered graph. ST-DRN and K3-DRN, owing to very few communication links, offer higher energy efficiency yet notably poor QoS (See Figure 7.5(a) and 7.5(b)), making them unsuitable for an effective DRN topology. Note that K5-DRN, despite having twice the number of communication links to that of Rand-DRN, offers better energy efficiency, thanks to its significantly lower network diameter. The Orig-DRN, due to its very high communication links, yields the worst energy efficiency. It is interesting to note that Bio-DRN is the only DRN topology that excels in energy efficiency and QoS (i.e., packet delivery and network latency), due to fewer communication links (or low graph density), and lower network diameter.

In order to analyze the scalability of Bio-DRN, we evaluate it against other topologies for varying graph orders with 3, 5 and 7 PoIs. Figures 7.6(b), 7.6(c), and 7.6(d) show that the energy efficiency, PDR and latency results for varying graph orders are consistent with those of previous results with 5 PoIs.

7.3.2.2. Robustness analysis. Here we utilize the snapshots of all network topologies (i.e., Bio-DRN, ST-DRN etc.) corresponding to every one hour (each capturing 2% node failure), and analyze the network robustness in terms of the following robustness metrics: (i) *Motif Count*, and (ii) *Path Count*.

7.3.2.2.1. Motif Count. We have discussed that FFL motifs render topological robustness to TRN by creating multiple paths. Figure 7.5(d) shows that Bio-DRN preserves high number of TRN motifs, possessing around 2 and 3 times the motif count compared to that of standard DRN topology with approximately same (Rand-DRN and K3-DRN) and half (K5-DRN) graph density, respectively (see Table 7.1). Evidently, the motif count in ST-DRN is 0 due to the absence of triangles, whereas the motif count of Orig-DRN is the highest because it is the densest DRN topology.

7.3.2.2.2. Path Count. Multiplicity of paths from any survivor node to the CC is an effective measure of network robustness as it ensures communication in events of failures. As shown in Figure 7.6(a), the path count in Bio-DRN is about 35, 40, and 6 times that of Rand-DRN, K3-DRN, and K5-DRN, respectively, notwithstanding node failures. Orig-DRN and ST-DRN exhibit the highest and lowest path count, again due to their high and low graph densities, respectively.

Analysis. Combining the results of robustness, energy efficiency and QoS, we infer that Bio-DRN is a promising approach for the design of the energy-efficient yet robust DRN topology, while ensuring desired QoS requirements.

7.4. INFERENCES

In this section, we designed an energy-efficient and robust DRN topology, termed *Bio-DRN*, which is inspired from a biological network of living organisms, termed transcriptional regulatory networks (TRNs). We formulated Bio-DRN topology construction as an integer linear programming optimization problem and prove its NP-hardness. We then proposed a polynomial time heuristic that constructs a Bio-DRN topology as a common subgraph of the Orig-DRN and TRN topologies, by exploiting the structural similarity between the genes and DRN components. Our simulation experiments on a real-disaster prone region, Bhaktapur, Nepal showed that Bio-DRN retains the topological properties of TRN. Furthermore, compared to other topologies, Bio-DRN exhibits a good balance between energy efficiency and network robustness against node failures, while ensuring timely data delivery. In the future, we shall explore designing faster heuristics for generating Bio-DRN topology that ensure optimal trade-off between the objectives of energy efficiency and robustness.

8. DATA TRANSFER FRAMEWORK OVER FOG COMPUTING PLATFORMS IN MOBILE CROWDSENSING

Urban areas are usually densely populated with few thousands of smartphone users spread across different geographic regions. Consequently, transferring and managing data from a large group of users is a bottleneck for both the underlying network (consisting of state-of-the-art Wi-Fi access points, gateways, routers, etc.) as well as the mobile crowdsensing (MCS) platform. Thus, we envision a MCS platform that deploys a fog computing framework across any smart city to facilitate efficient data transfer. Typically, microservers like notebooks, laptops, etc. are used as fog devices.

In traditional settings, the task data generated by mobile devices are delivered to a base station (hosting the MCS platform) in a multi-hop fashion. In each hop, the nearest fog device forwards the data to another fog, until the data is transferred via the gateway fog device. This results in greater message delay, and also keeps multiple fog nodes engaged majority of the time causing higher energy dissipation. Moreover, as all fog devices will not be forwarding a uniform number of requests, an issue of scalability may also arise. On the contrary, smart city applications have to be sustainable in terms of energy efficiency and scalability. Often the fog devices may be deployed at locations remote to the MCS platform and their limited energy gets dissipated while communicating the sensed data via wireless communication technologies, such as 3G/4G/LTE, Wi-Fi, etc. Furthermore, it may not always be feasible to replenish their batteries nor replace them with fully-charged devices on-the-fly. Therefore, it is critical to design efficient data transfer framework for fog-based smart city applications.

Considering the need to come up with energy efficient networking solutions in smart city applications, we propose a bio-inspired collaborative data transfer framework through mobile crowdsensing over fog computing platforms. Specifically, we make the following contributions in this work:

- Identify clusters of densely connected fog devices, while optimizing cluster modularity.
- Utilize a TRN-based mapping strategy to design sparse yet robust fog network topologies.
- Formulate a fitness function to identify gateway fog nodes.
- Design a weighted preference to find mobile group owners to enable collaborative sensing through autonomous Wi-Fi direct mode.
- Perform simulation based experiments to study *bioMCS* in terms of *energy efficiency*, *robustness*, *load balancing* and *data delivery*.

8.1. SYSTEM MODEL

This work proposes a framework for efficient data delivery in mobile crowdsensing (MCS)-based smart city applications. We consider an urban space which has thousands of users with smart hand-held devices (e.g., smartphones, tablets, wearables, etc.). MCS platforms leverage the sensing capabilities of the mobile users to collect rich environmental information for providing services in various domains, such as environmental monitoring, social networking, healthcare, transportation and safety. We envision a data collection architecture consisting of several fog devices deployed across the urban space. Moreover, mobile users who are spatially close to one another can leverage device-to-device proximity based sensing to save energy, network bandwidth, and reduce data duplicity. Such architecture will also enable the MCS platform to reach out to remotely located mobile users, thus enhancing the data acquisition.

Figure 8.1 depicts our system model that captures an urban space, equipped with multiple fog devices $f_1, f_2, f_3, \cdots$ and thousands of mobile device users $\mathcal{M} = \{d_1, d_2, d_3, \cdots\}$. The system model also consists of a *base station* which is hosting the MCS platform. The

MCS platform assigns various sensing tasks to the mobile device owners which are participatory in nature and explicitly requires users to sense and report against the assigned tasks. The major components of our framework are as follows:

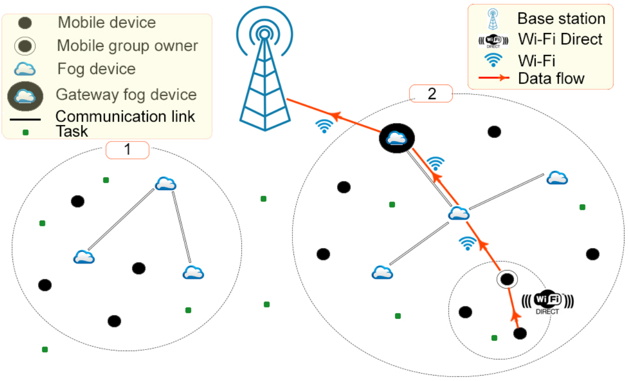


Figure 8.1. System model.

1. Fog device: A fog device f_i is a rechargeable, energy-constrained node with routing, processing, and storage capabilities. Typically they are edge devices, such as cloudlets, capable to location-aware and low latency computing. They are equipped with data aggregation functions to reduce the amount of information delivered to the remote back-end servers. Multiple fog devices are deployed across geographically dispersed locations, which can be divided into clusters based on spatial proximity. Within a cluster, one of the fog devices is chosen as gateway based on its connectivity and residual energy. A fog device serves as an intermediary between the mobile devices and the base station. We define an *original fog topology* $\mathcal{F}$, where nodes are fogs and two-way communication link exists between a pair of fog nodes if they are in range R_f.

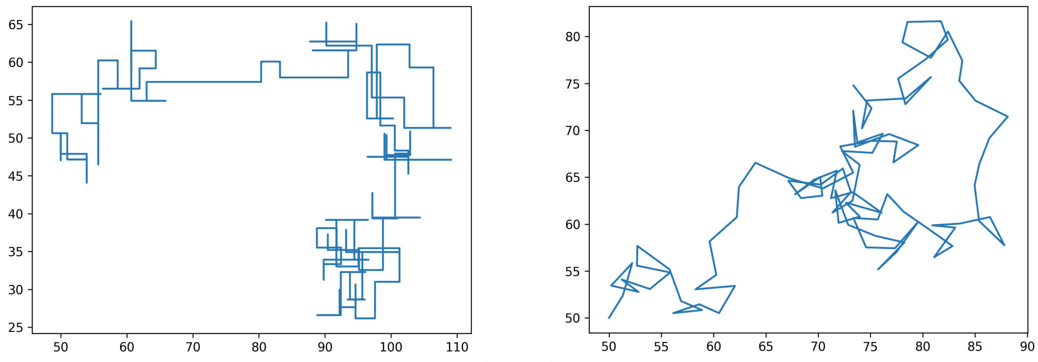
2. Mobile device: A mobile device $m_j \in \mathcal{M}$ is a energy-constrained, handheld device viz., smartphone, notebook, tablet computers, wearables, etc., which a user/participant will possess. These devices change their positions with time and have installed the application developed and managed by the MCS platform. The platform pushes crowdsensing tasks through the application, which also provides interfaces for submitting data. Each mobile device will have a preference score which combines its *activeness* in using the MCS application, *promptness* in accepting and executing the tasks, and residual energy. As far as participation in the MCS tasks, the operations of a mobile device switch between two roles: (i) *Wi-Fi direct Group Owner (GO)*: Under this role, the mobile device is selected to be most suitable to establish a group; (ii) *Peer nodes*: In this role, the mobile node voluntarily performs Wi-Fi direct scanning to discover the *GO* and get attached to the group.

3. Mobility models: The mobility of the mobile devices follow two similar random walk mobility models. In both the methods, the mobile nodes move randomly across the deployment region with varying speed. Below are the brief descriptions of the two variants: *a. Random walk mobility model:* In this model [156], at each time epoch, a mobile device $m_j \in \mathcal{M}$ at location (x_t^u, y_t^u) randomly chooses a direction θ $(0 \leq \theta \leq 2\pi)$ and random speed v_t $(v_{min} \leq v_t \leq v_{max})$. Its new location at time $t + 1$ is:

$$(x_{t+1}^u, y_{t+1}^u) = (x_t^u + v_t \times \cos(\theta), y_t^u + v_t \times \sin(\theta)) \tag{8.1}$$

If the device reaches a grid boundary, it is reflected with an angle determined by the incoming direction. We also consider a special case of random walk mobility model in which the node mobility is restricted to left, right, top and bottom. *b. Random waypoint mobility model:* A node halts in one location for a brief period (i.e., a pause time). It then chooses a random destination as well as a speed v_t $(v_{min} \leq v_t \leq v_{max})$. It then travels towards the newly chosen destination at the selected speed. Upon arrival, the device takes a pause before repeating the same step. Note that random waypoint model converges to random walk mobility model when the pause time is 0 [157].

Figures 8.2(a) and 8.2(b) show the random walk and waypoint mobility on a single device for a duration of 200 min., speed v_t in range $[2, 4]$ m/min. The pause time for the waypoint model is 2 min.



(a)

(b)

Figure 8.2. Mobility model. Mobility of a single device for 200 min. and speed v_t ranging between $[2, 4]$ m/min. for (a) Random walk mobility model (b) Waypoint mobility with pause time = 2 min.

4. Task: Let a set of tasks $\mathcal{T} = \{t_1, t_2, t_3, \cdots\}$ are generated by the MCS platform within the urban space at a particular time instant. Any task $t_k \in \mathcal{T}$ (denoted by green dot in Figure 8.1) is an alert which gets triggered by the MCS platform at a location within the urban space. Few examples of MCS tasks are giving ratings, uploading photos and sharing details of places of interest, and so on. In response to tasks, the mobile device users submit reports to the MCS platform. Each task is associated with an identifier, a geographical location information and deadline within which it can be finished.

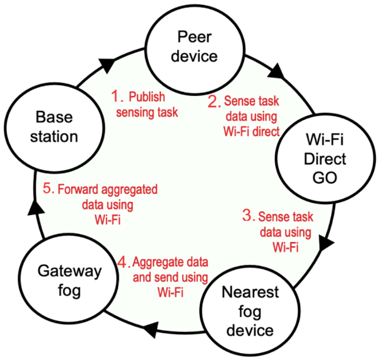

Peer device
Base station
Wi-Fi Direct GO
Nearest fog device
Gateway fog
1. Publish sensing task
2. Sense task data using Wi-Fi direct
3. Sense task data using Wi-Fi
4. Aggregate data and send using Wi-Fi
5. Forward aggregated data using Wi-Fi

Figure 8.3. Communication sequence after every time epoch.

5. Base station: A base station B hosts the MCS platform which is remotely located from the urban space (refer Figure 8.1). Its communication range (R_B) is much higher compared to that of fog devices, and it uses a plethora of communication protocols and the backbone Wireless LAN network for bidirectional message exchange with the devices. The base station is responsible for generating tasks at different geographical regions and receiving reports from the mobile devices via the gateway.

We introduce the concept of *time epoch*, of duration τ, to delineate different processes taking place periodically in each cycle of data transfer. It is a configurable parameter, whose duration depends on the frequency of data samples collected by the base station. Specifically, after every time epoch, the following processes take place: (1) base station publishes sensing tasks; (2) peers, in mobile groups after sensing the tasks, transfer information to the GOs; (3) GOs identify the nearest fogs and transfer the collected data to them; (4) fog devices transfer the aggregated and non-redundant task information to gateway fog; (5) gateway sends task data to base station (as shown in Figure 8.3).

8.2. BIOMCS FRAMEWORK

In this section, we cover the different facets of *bioMCS* framework. First, we propose a bio-inspired mapping strategy that employs TRN to construct a sparse yet robust fog network topology. We then elucidate the energy efficient data transfer from mobile peers to base station.

8.2.1. Bio-inspired Hierarchical Mapping. We propose the hierarchical mapping strategy that is based on our previously proposed algorithm for design of robust WSNs [45]. We first partition the fog network $\mathcal{F}$ into clusters of densely connected devices and apply the mapping to each cluster. The steps in the hierarchical mapping approach are discussed as follows:

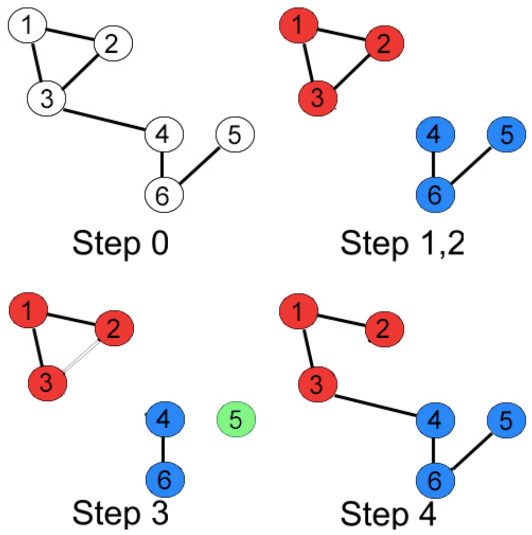


Figure 8.4. Bio-inspired mapping. (1) Original fog network $\mathcal{F}$ (2) Nodes in $\mathcal{F}$ are grouped into 2 clusters (3) Mapping algorithm is applied to each cluster: edge $e(3, 2)$ and node 5 (colored green) are unmapped (4) Mapped fog network $\mathcal{F}^m$.

8.2.1.1. Step 1: Graph partitioning. In this step, we apply the *agglomerative hierarchical clustering algorithm* (implemented using the Python Scikit Learn library [158]) to partition the *original fog network* $\mathcal{F}$ into disjoint clusters. At the outset, each fog node f_i is a standalone cluster; thus, we have clusters $c_1, c_2, \cdots, c_{|V(\mathcal{F})|}$ in the system. In each iteration of the clustering, two clusters c_i and c_j are combined if their respective member nodes u, v have the smallest euclidean distance i.e., $min\{d(u, v) : u \in c_i, v \in c_j\}$. We iteratively generate $|V(\mathcal{F})|$, $|V(\mathcal{F})| - 1, \cdots, 2$ clusters and select the cluster configuration of the highest quality, determined by a metric called *modularity* (defined below). In Figure 8.4 we show 2 clusters of $\mathcal{F}$.

Modularity score: *Modularity is defined as the fraction of the edges that fall within the given groups minus the expected fraction if edges were distributed at random* [159]. Graphs with high modularity possess high inter-cluster distance and low intra-cluster distance. Given a graph $G(V, E)$, we calculate modularity using the equation,

$$M = \frac{1}{2|E|}\sum_{u,v\in V}(\mathbb{A}_{u,v} - \frac{k_u * k_v}{2|E|})\delta(\mathbb{C}, u, v) \tag{8.2}$$

Here (i) $\mathbb{A}$ is the adjacency matrix of G, (ii) χ is a list such that χ_u holds cluster id for node u and (iii) delta function $\delta(C, u, v)$ returns 1 if $\chi_u = \chi_v$ and 0 otherwise. (iv) k_u is the degree of node u.

8.2.1.2. Step 2: Preprocessing for TRN-based mapping. Once $\mathcal{F}$ has been partitioned into κ clusters $c_1, c_2, \cdots, c_\kappa$, the inter-cluster edges are removed. We remove all edges $ie = \{e(u, v) \in E(\mathcal{F}) | \chi_u \neq \chi_v\}$. Next, for each cluster c_i, we generate an induced subgraph $\mathcal{F}_i'$ comprising nodes of that cluster. In the next step the TRN mapping algorithm is applied to each $\mathcal{F}_i'$.

8.2.1.3. Step 3: TRN based mapping algorithm. We apply the mapping algorithm to induced fog network in each cluster to generate a robust yet sparse fog network topology within each cluster.

8.2.1.4. Step 4: Graph connectivity. The hierarchical mapping algorithm generates the mapped fog topology $\mathcal{F}^m$ by combining $\mathcal{F}_i^m$s by graph union (for all clusters i), and restores the intercluster edges ie. However, the above steps may still lead to a disconnected mapped fog topology. Thus, to ensure connectivity, we employ a *connect* function.

This *connect* function works as follows. Assume there are n_c (disconnected) components in G_{mw}. For each pair of components, we add an unmapped edge from $\mathcal{F}$ that connects the two components. If no such edge exists, we include the shortest path from $\mathcal{F}$ that connects the two components. This may entail the inclusion of unmapped nodes to $\mathcal{F}^m$. As an example, consider $\mathcal{F}$ (shown in Figure 8.4(1)). Note that node 5 (labeled green) is not mapped, leading to a disconnected graph. In order to obtain a connected mapped topology $\mathcal{F}^m$, we restore unmapped edge $e(5, 6)$ to $\mathcal{F}^m$. Therefore, this step transforms the mapped topology into a connected topology ensuring potential communication among different clusters (although in the present architecture we do not make use of the intercluster links).

8.2.2. Energy Efficient Data Dissemination. We dedicate remaining part of this section to discuss the other aspect of *bioMCS* i.e., energy efficient transfer of task data by the mobile devices to the fog devices. We also discuss how the task data is routed through the mapped fog topology to the base station.

8.2.2.1. Gateway fog selection. Each cluster possesses a gateway fog (GF) selected from among the fog nodes in that cluster. The GF is responsible for collecting all the task information sensed by the mobile devices in a cluster and forwarding it to the base station. Note that in every l^{th} cluster, the GF device is re-selected by the base station after every time epoch, on the basis of a fitness score, computed as follows:

$$\rho_f = w_f \times \frac{d_f}{\max_{f':C_{f'}=l} d_{f'}} + (1 - w_f) \times \frac{e_f}{\max_{f':C_{f'}=l} e_{f'}} \tag{8.3}$$

In the above equation, we estimate the fitness of fog device f based on the combined score of normalized degree of connectivity (d_f) with other fogs in mapped fog network, and residual energy (e_f). High degree and residual energy ensure that the GF is capable of interacting with and acquiring task information from a large number of fogs. Note that w_f ($0 \leq w_f \leq 1$) is the parameter that controls the weight attached to degree and residual energy.

8.2.2.2. Group owner selection and group formation. Recall from the discussion in Section 8.1, tasks are generated by the MCS platform at random locations in the urban space. For each such task, the GF, which periodically scans for mobile devices in the vicinity of its cluster, selects the fittest mobile device in sensing radius (R_t) of the task. The node with the highest preference, calculated using Eq. 8.4, is selected as the group owner (GO).

$$\begin{aligned}\rho_m = w_a \times \frac{a_m}{\max_{m':C_{m'}=l} d_{m'}} + w_p \times \frac{p_m}{\max_{m':C_{m'}=l} p_{m'}} + \\ w_e \times \frac{e_m}{\max_{m':C_{m'}=l} e_{m'}}\end{aligned} \tag{8.4}$$

In the above equation, similar to GF fitness, we gauge the preference of mobile device m based on the weighted sum of normalized values of three parameters activeness (a_m), promptness (p_m) and residual energy (e_m). Out of these three parameters a_m is in range $[1,4]$ (where 4 denotes maximum activeness); p_m is reset after every η time units to exclusively consider recent response of a device to tasks. The weights w_a, w_p and w_e sum up to 1. It is noteworthy that there can be several GOs in a cluster, depending on the frequency and location of tasks. Each peer node scans for GOs in its communication range R_m and joins its *mobile group*.

8.2.2.3. Task sensing and forwarding task information. Once the mobile devices accept the invitation to join the respective mobile groups, they are instructed to sense the tasks. If a mobile peer m accepts the task sensing invitation and successfully senses the tasks, its promptness score p_m is incremented, otherwise it stays the same. We assume that device m has equal probability of accepting task sensing invitation as refusing it. The sensed task information is forwarded to the GO, who transfers it to the nearest fog device. In case a peer device has no GO at a certain time epoch, it directly transfers the task information to the nearest fog device. This step is particularly significant in the context of any data transfer framework because it is imperative to ensure that no amount of task information is lost due in the process of enforcing protocol.

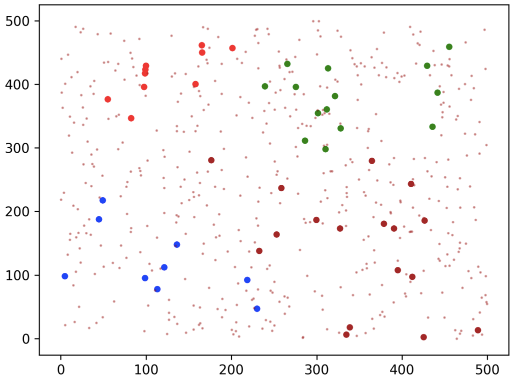


Figure 8.5. Deployment of mobile and fog devices. 500 mobile devices (small brown dots) and 50 fog devices (larger circles) deployed in a region of 500 × 500 sq. m. Different colours on larger circles indicate different clusters.

8.2.2.4. Task information filtering and transfer to base station. A fog device that acquire the task information from the GOs undertakes *filtering of task information*. In this step, the fog device scans all the task information in its memory by its task ID and eliminates all the redundant task information. Following this, the task information is forwarded to the GF, from which it is sent to the base station.

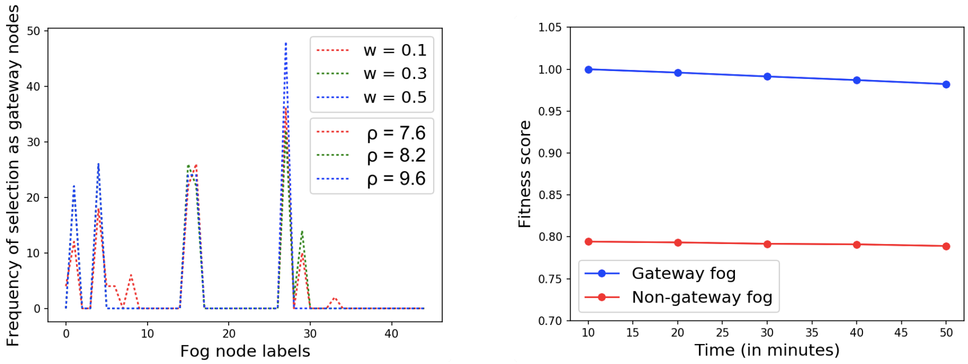
8.3. EXPERIMENTAL RESULTS

For our experiments we develop a customized discrete event simulator based on Python Simpy library [144]. We use *S. cerevisiae* TRN (4441 nodes and 12873 edges) for hierarchical mapping. We first discuss the results on robustness and energy efficiency rendered by TRN-based mapping. Then, with regard to the fog devices, we carry out experiments on the graph properties of mapped fog network, task filtering at nearest fog and selection of gateway fog (GF) nodes. Finally, we study the mobility of peer devices and different facets of group owner (GO) selection.

Simulation Setting and Parameters: We simulate a deployment region of 500×500 sq. m., that consists of a base station, 50 fog devices and 500 mobile devices. The communication range of the fog devices and mobile devices are 40 and 20 m., respectively. The base station is Wi-Fi enabled and possesses a communication range of 500 m. Figure 8.5 shows the snapshot of a graphical representation of the deployment space, where the small and large circles are mobile devices and fog nodes, respectively; different colours of fog nodes suggest that they belong in different clusters.

The system generates tasks at random locations and the frequency of tasks follow exponential distribution with mean 20. The simulation duration is 50 time units.

8.3.1. Properties of Gateway Fogs (GFs). In these experiments we shall evaluate the behavior of the GFs w.r.t. the fitness weights and fitness scores.



(a) (b)

Figure 8.6. Properties of Gateway Fog. (a) Effect of fitness score weight on the selection of GFs (b) Fitness of GFs vs. non-GFs.

8.3.1.1. Effect of fitness weight w_f on GF selection. Recall that the fitness score for any fog device has two components: connectivity (i.e. degree) and residual energy. We regulate the fitness weight w_f and study its effects on the selection of gateway devices. Figure 8.6(a) shows the frequency of a node to be selected as gateway for three possible values for w_f $(0.1, 0.3, 0.5)$. When $w_f = 0.1$, the fitness score strongly prefers residual energy over degree; since residual energy of a fog device diminishes over time, different fogs get selected as GFs. This uniformity is reflected in the low standard deviation in device selection (ρ). Conversely, for a $w_f = 0.5$, the fitness score gives high weightage to degree, which can only change when neighbor fogs run out of energy. Thus, few nodes with high connectivity are preferred as GFs, resulting in high ρ.

8.3.1.2. Fitness of GF devices. From the discussion in Section 8.2.2, we know that the nodes with the highest fitness get selected as GFs in each cluster, which channel the task information to the base station. For a fixed fitness weight $w_f = 0.5$, we compare the average fitness score of GFs versus those of non-GFs, over time.

Figure 8.6(b) shows that the average fitness score of GFs and non-GFs start at 1.0 and 0.8 and drop marginally over time. Overall, the fog devices with the highest fitness get selected as GFs.

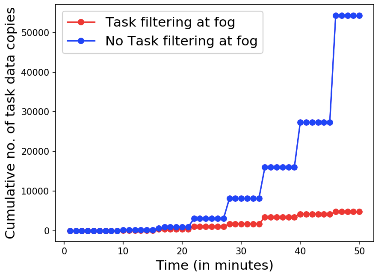
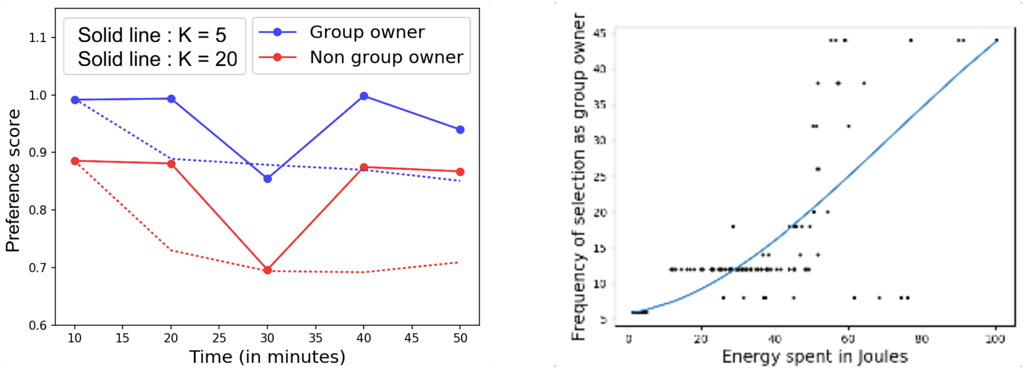


Figure 8.7. Task filtering vs. no filtering at fog.

8.3.2. Effect of Data Filtering at the Fog Device. We discuss in Section 8.2.2.4, the task information sent by the group owners to the nearest fog device undergo a filtering process where the redundant task information is eliminated. In this experiment, we compare the number of task data copies generated under conditions of filtering and no filtering at the nearest fog device.

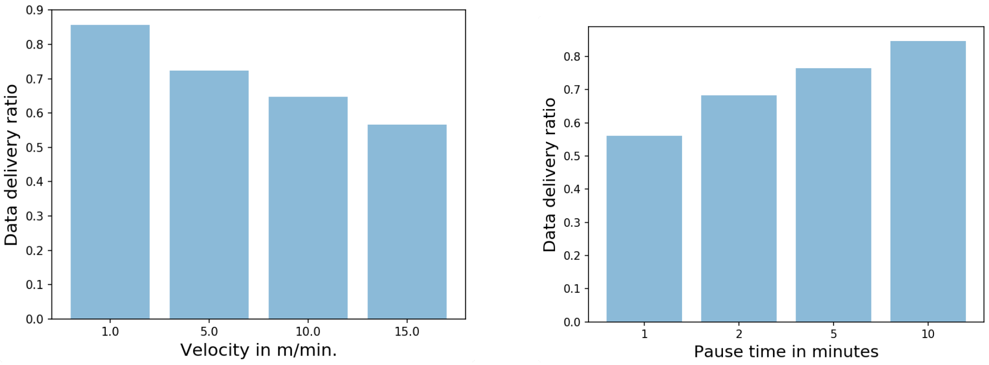


(a)

(b)

Figure 8.8. Effect on GO. (a) Preference score of GO vs non-GO mobile device (b) Energy consumption of GO.

Figure 8.7 shows that the gap in the curves for task data copies for filtering and no filtering widen over time. At the end of the simulation, the number of data copies produced with no filtering is nearly ten times that of the filtering condition.



(a)

(b)

Figure 8.9. Effect on data delivery. (a) Velocity and (b) Pause time.

8.3.3. Effect of Mobility on Data Delivery. We analyze the effect of the movement of mobile devices on the overall data dissemination (or delivery) of the system. We define data delivery rate as the number of tasks generated by the system to the number of tasks reported at base station. We vary the two parameters of random waypoint mobility model: *velocity of mobile devices* (v) and *pause time* (π), and measure the data delivery rate.

Figure 8.9(a) shows that for a constant π of 2 minutes, the increase in v of mobile node causes a drop in data delivery ratio at the base station. Similarly, for a constant v of 2 m/min., increase in π enhances the data delivery rate (Figure 8.9(b)). In both cases, high mobility among mobile nodes leads to poor data delivery rate. This is because high mobility often causes the devices to move out of sensing range of the task assigned to it, leading to low data delivery.

8.3.4. Properties of Mobile Group Owners (*GO*s). In the subsequent experiments we analyze the fitness and energy consumption of *GO*s from different standpoints.

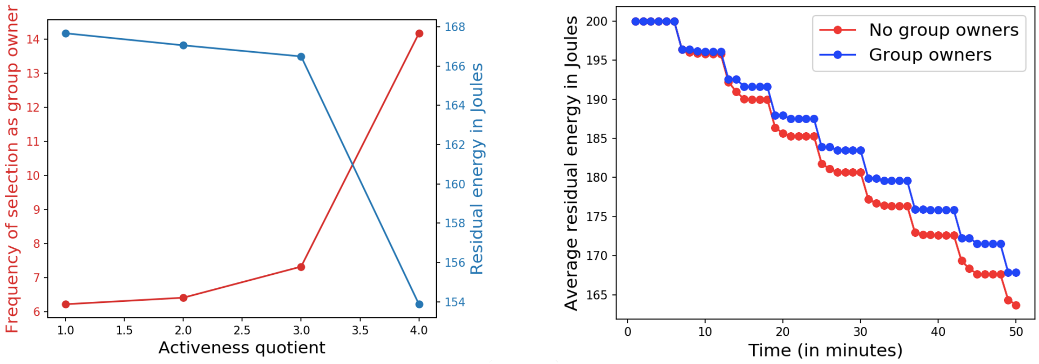
8.3.4.1. Preference of mobile devices. We now analyze the preference score of *GO*s as compared to the non-*GO*s over time. Recall that preference of mobile devices is a weighted sum of activeness, promptness and residual energy. Also, the promptness quotient of each mobile device is reset to 0 after every η time epochs. We consider activeness, promptness and residual energy weights $w_1 = 0.4, w_2 = 0.3, w_3 = 0.3$, respectively; we also study two distinct $\eta = 5, 20$.

Figure 8.8(a) shows that for both cases of $\eta = 5$ as well as $\eta = 20$, the preference score of the *GO*s is significantly higher than the non *GO* mobile devices. Also, for $\eta = 5$, the promptness is reset to 0 frequently, and as a consequence, the preference rises and falls more frequently, than in case of $\eta = 20$.

8.3.4.2. Energy consumption by *GO*s. We study the relationship between the selection of mobile nodes as *GO*s and energy consumed for forwarding task information to nearest fog. Figure 8.8(b) shows the strong correlation between the frequency of selection of a node as *GO* and its energy spent. Non-linear curve fitting (with degree 3) on the data points (each signifying a mobile device) shows that mobile group owners tend to consume more communication energy.

8.3.4.3. *GO* selection and residual energy of active devices. In Section 8.1, we define activeness score of a node on a scale of $1 - 4$, where 4 stands for highest activeness. Figure 8.10(a) shows the residual energy and frequency of selection as *GO*s on the basis of their activeness score. The result corroborates that active mobile nodes are chosen as *GO*s and consequently consume high communication energy for forwarding the task information to the nearest fogs.

8.3.4.4. Effect of *GO* on energy efficiency. We analyze the role of *GO*s in rendering energy efficiency to the system. We measure the average residual energy of the mobile devices under conditions of *GO*s and no *GO*s. Figure 8.10(b) shows that the presence of *GO*s lead to notably low consumption of communication energy. This is because, in the absence of *GO*s, the peers consume higher energy to transmit task information using Wi-Fi connectivity as opposed to the Wi-Fi direct used by peer devices to interact with *GO*s.



(a)

(b)

Figure 8.10. Group owner selection and energy consumption. (a) Variation of GO selection and residual energy with activeness of mobile devices (b) Comparison of GO vs. no GO on energy efficiency.

8.4. INFERENCE

In this work we proposed a bio-inspired transcriptional regulatory network (TRN) based data transfer framework *bioMCS* to facilitate energy efficient task data collection through mobile crowdsensing. The framework enables mobile devices to undergo Wi-Fi direct-based collaborative sensing and save considerable battery energy while transferring task information to the base station. Moreover, we assumed that the framework is deployed over a fog computing platform, where the fog devices are used to select Wi-Fi direct group owners, aggregate data from them, and forward it to the MCS platform hosted in the base station. *bioMCS* first ensures even distribution of task load by partitioning the fog network into clusters, and then applies the bio-inspired hierarchical mapping algorithm to generate sparser fog topology that inherits the topological robustness of TRNs. We evaluate our framework through extensive simulation-based experiments and demonstrate that the *bioMCS* framework achieves better energy and network efficiency compared to individual user-centric data transfer mechanism.

There is one aspect of the proposed *bioMCS* framework that we are trying to address through our current research. From our discussion in Section 8.2.1 we are aware of the existence of the inter-cluster edges. Although, the purpose of the inter-cluster edges is not well-defined in this work, we intuit that they may help in addressing the challenges with decentralized data transfer between mobile nodes across clusters. Specifically, the decentralized framework will support the mobile users to access information of a task taking place at different clusters.

9. ONGOING RESEARCH: A COMPUTATIONAL FRAMEWORK TO IDENTIFY MINIMAL DRUGGABLE TARGETS IN TRANSCRIPTIONAL NETWORKS

Specific genes play a crucial role in disease biology by affecting signaling pathways and different cancer types. Taking inspiration from a similar framework designed to study miRNA-miRNA regulation [160], we propose a computational framework to identify the set of druggable spreaders in TRNs that can influence (i.e. positively or negatively activate) specific genes. Our work consists of two phases: (1) learn the edge weights that signify the strength of TF-gene regulation, and (2) apply influence diffusion mechanisms to identify the minimum druggable targets.

9.1. LEARNING EDGE WEIGHTS

We discussed in Section 1.3, TRNs are signed networks. In this section, we attempt the find the weights on the signed edges; the weights range between -1.0 signifying maximum negative regulation to 1.0 signifying maximum positive regulation.

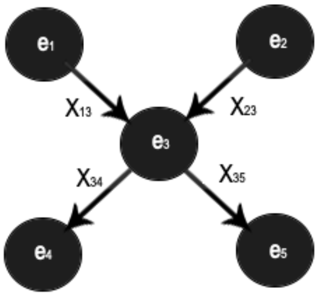


Figure 9.1. Schematic representation of organization of motif central nodes within tier 2 of a TRN.

To achieve the above objective, we consider the expression score for each TRN node as its node weight and assume the change in expression score will flow throughout the TRN via the directed links. Specifically, a node having a very high expression score would propagate effect on its neighboring miRNAs. Thus, we define two ground rules:

Variables: Let $X_{u,v}$ (ranging between -1.0 to 1.0) be the directed flow of influence from node u to node v, where $u, v \in V$. Given any node v, e_v be the fold-change expression, and s_v^i and s_v^o be the incoming and outgoing slack variables (discussed hereafter).

To support our assumptions, we define the following two constraints:

- The cumulative sum of products of **i**ncoming edge-weights and corresponding expression scores of parent nodes would exceed the expression score of the target node v by a slack variable (s_v^i in Constraint 9.4).
- The sum of node **o**utgoing edge-weights of any node v can exceed its expression score within a slack amount (s_v^o in Constraint 9.5).

To clarify the above rules, consider a toy network of 5 nodes (Figure 9.1). If we focus on the central node with expression score X_5, then the collective influx of expression into the node = $\sum_{i=1,2} e_i X_{i,j}$ and its outflow = $\sum_{i=4,5} e_i X_{j,i}$. Then,

$$\sum_{i=1,2} e_i X_{i,j} \approx e_3 \tag{9.1}$$

$$\sum_{i=4,5} e_i X_{j,i} \approx e_3 \tag{9.2}$$

The objective function is to minimize the sum of slack variables s_i and s_o over all the nodes (Expression 9.3).

$$\textbf{Min.} \sum_{v \in V} |s_v^i| + \sum_{v \in V} |s_v^o| \tag{9.3}$$

$$\textbf{s.t.} \sum_{u \in V} e_u * X_{u,v} + s_v^i = e_v \qquad \forall v \in V \tag{9.4}$$

$$\sum_{u \in V} e_v * X_{v,u} + s_v^o = e_v \qquad \forall v \in V \tag{9.5}$$

Range of $X_{u,v}$: Any directed edge $e(u, v)$ can have one of three possible states: activation, repression or unknown. If the state is activation $e(u, v)$ lies in range $(0, 1.0)$; if it is a repression it resides in the range $(-1.0, 0.0)$; otherwise it can belong in range $(-1.0, 1.0)$ allowing the optimizer determine the nature of regulation.

Linear and nonlinear formulations: For our initial dataset, we only have the expression values of 2103 out of the 2862 nodes in Human TRN. Therefore, the above optimization formulation becomes nonlinear in nature. In order to aid computations for the complete Human TRN, we convert the formulation to linear. Given known and unknown expression values $\mathcal{E}_{known}$ and $\mathcal{E}_{unknown}$, respectively, we set the expression value of each $v \in \mathcal{E}_{unknown}$ to the mean of expression value of known nodes i.e., $e_v = \frac{\sum_{u \in \mathcal{E}_{known}} e_u}{|\mathcal{E}_{known}|}$ $(\forall v \in \mathcal{E}_{unknown})$.

Drawbacks: Although these assumptions can be expressed in the form of a linear and nonlinear optimization formulation, leading to convenient solutions, it has the following drawbacks.

- Our proposed method relies heavily on the exact values of expression of the nodes. Since different TRN nodes possess different expression values under different circumstances, there is likely to be a high variability in expression values across different scenarios.
- The TRNs of human and mouse are incomplete i.e. all the Transcription Factor-gene interactions are not documented. Therefore, it is difficult to arrive at accurate edge weights.

- The TRNs are expected to be dynamic networks with changing node weights and edge relationships. Since our analysis is based on snapshots of TRNs, it only reflects the steady-state values.

9.2. INFLUENCE DIFFUSION

With the knowledge of the edge weights in transcriptional networks, we now try to devise a computational framework that will address the following question:

9.2.1. Problem. Given a set of ordered pair of target TRN nodes and desired activation status (i.e. positive (+) or negative (−)) $Q = \{(q_1, < +/- >), (q_2, < +/- >), ...\}$, we attempt to come up with k causal nodes that will maximize activation of the target nodes (with the correct sign) and minimize activation of the non-target nodes. (Note that all the nodes in TRN not in target set are considered to possess activation status $None$).

9.2.1.1. Approach. Our proposed framework is based on the well-studied influence diffusion (or maximization) mechanism proposed by Kempe et al. in his seminal work [161]. Kempe defined the influence diffusion problem as choosing a set of individuals to target for initial activation, such that the cascade beginning with this active set (or called *causal* or *seed* set) is as large as possible in expectation. He went on to prove that the problem is sub-modular in nature and the greedy algorithm is bounded by an approximation with a factor of $1 - 1/e$. In our context, influence diffusion is applied on a signed network (i.e. TRN) [162] with a specific target set.

9.2.1.2. Algorithmic step. We model our solution on the Independent Cascade (IC) influence maximization mechanism where the method is to begin with an empty seed set and then iteratively add nodes with the maximum spread (averaged out over several iterations) in the network. The process for influence spread is simple: if a random number $r \in [0, 1]$ is less than or equal to $X_{i,j}$, the activation status of node j is the product of the activation status of node i and $sign(X_{i,j})$.

Let us now discuss the key considerations in our proposed solution:

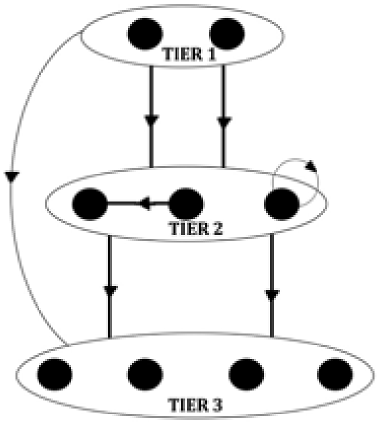


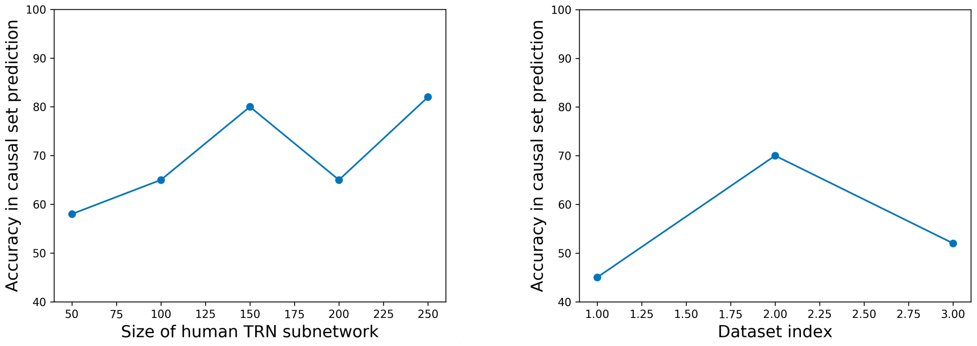
Figure 9.2. Three tier topological characterization.

1. *Spreaders belongs in tiers 1 and 2:* Recall from our discussion on three tier topology in Section 3, all the spreaders in the TRN topology belong in tiers 1 and 2 (containing nodes with all the outgoing edges) (Figure 9.2). Since tiers 1 and 2 account for approximately 10% TRN nodes, we can restrict our search of causal nodes to to tiers 1 and 2. This makes the greedy search significantly more computationally feasible.

2. *Gain function to gauge impact on target:* In order to analyze how the chosen seed set influences the target, we have (so far) come up with two simple cost/gain functions. Below we describe two simple cost functions.

   - **Scoring 1:** Let TS be the nodes in the actual target set that is correctly identified (with the correct activation status) in the predicted target set. Similarly, let AN be the actual set of non-target nodes and ON be the observed set of non-target nodes. The cost function is calculated as $\frac{|TS|}{|Q|} * w_1 + \frac{|AN|}{|ON|} * w_2$, where $0 < w_i < 1$ ($\sum w_i = 1$) is the weighing factor.

   - **Scoring 2:** Let TS_0 be the intersection between the predicted and actual target set. Similarly, let TS_N be the number of nodes in TS_N identified with the wrong sign. Then, the cost function equals $\frac{|TS_0|}{|Q|} - \frac{|TS_N|}{|TS_0|}$.



(a)

(b)

Figure 9.3. Prediction accuracy. (a) Cross-validation and (b) Real data.

9.2.2. Cross-validation. This is standard procedure we follow to analyze the effectiveness of the proposed heuristic. In this, we start with a set of seed nodes and run the forward signed influence diffusion mechanism to identify the target set Q. Then, we follow the usual backward influence diffusion to identify the seed set from the observed targets.

9.3. RESULTS

Although we have limited dataset at our disposal, we are currently generating synthetic datasets to validate and refine our proposed computational framework. We obtained the expression score values for human TRN from Expression Atlas [163]. The causal and targeted node datasets were obtained from [164].

9.3.1. Cross-validation. In this experiment, we apply the cross-validation approach on TRNs of different orders. Our results show that the accuracy in determination of the causal nodes range between $60 - 80\%$ (Figure 9.3(a)).

9.3.2. Real Dataset. Our experiments on real datasets show a slightly lower accuracy (ranging between $40 - 70\%$) (Figure 9.3(b)). The reason for this performance decline has been discussed in the next section.

9.3.3. Challenges. The proposed computational framework suffers from two interesting research challenges.

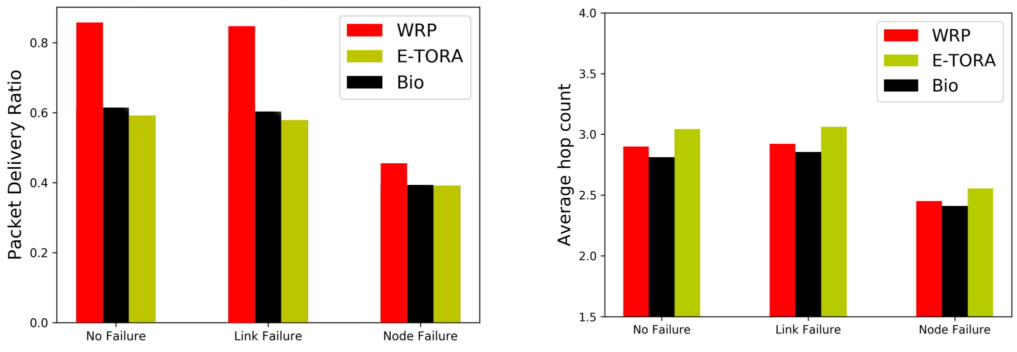
- We have discussed two possible cost functions in Section 9.2. Clearly, there could innumerable other variants of the same function. We observe that the accuracy in identification of the causal nodes depend heavily on the choice of cost function. The identification of the correct cost function itself poses several research questions.
- When working with the real dataset, our framework often identifies nodes which are not documented as causal nodes, but are their predecessors in the three tier topology. This results in lower accuracy. However, in reality, our observed causal nodes could well be considered to be the nodes responsible for the activation of the given target set. We are looking into possible ways to address this problem.
- The proposed problem solution can be shown to be non sub-modular in nature. The fact that we are unable to utilize certain known properties of submodularity, greatly restricts the efficacy of the computational framework.

10. CONCLUSION AND FUTURE DIRECTIONS

In this dissertation we study the topological properties of a biological network called Transcriptional Regulatory Network (TRN) based on a three tier topological characterization. We analyze the properties such as abundance of Feed Forward Loop (FFL) motifs, low graph density, scale free out-degree distribution. Our studies show that FFL motifs are responsible for the high clustering tendency of TRNs and play an essential role in signal transduction by creating communication pathways. Moreover, the motif central nodes are the most efficient spreaders of information in TRNs. In addition, we come up with a computational framework that identifies the most effective druggable targets in TRNs. Finally, we apply the TRNs in the design of robust and energy-efficient wireless sensor networks, disaster response networks and data transfer frameworks over IoT and fog computing platforms. Let us now go over the unexplored areas of TRNs which can motivate new research directions in the fields of communication and social network analysis.

10.1. BIO-INSPIRED ROUTING PROTOCOL

Our discussion in Section 5 shows that TRNs serve as an effective template for the design of robust static WSN topologies. TRN-based WSNs achieves fault-tolerance by maximizing FFL motifs. Additionally, our recent findings reveal that FFL motif central nodes are the most effective information forwarders in TRNs [165]. Based on these findings, we plan to investigate the possibility of dynamic bio-inspired routing protocols in a WSN setting using the *node motif centrality* (Eq. 4.1). We can deduce that in any given FFL role A motif centrality creates two independent paths between master regulator and regulated node; thus, a node with high role A FFL motif centrality will potentially present several communication pathways between any given source and sink node pairs.



(a)

(b)

Figure 10.1. Bio-inspired routing. Comparison of performance of WRP, E-TORA and Role A motif centrality based routing (a) Packet delivery ratio (b) Average hop count.

To verify our intuition, we devise a simple bio-inspired routing strategy wherein each node maintains information of the role A FFL motif centrality of the neighbor nodes and choose the node with highest role A motif centrality as its next hop. We compare our strategy with two standard routing protocols: (1) Wireless Routing Protocol (WRP) [166] which utilizes shortest path based schemes to calculate the minimum cost routes, and (2) Energy-aware Temporally Ordered Routing Algorithm (E-TORA) [167], which conserves energy by taking into consideration the level of power of each node and avoids using nodes with low residual energy. Our simulation experiment on a WSN of 50 nodes designed on a discrete event simulation environment of Python SimPy library [144] reveals that the packet delivery ratio (PDR) for WRP is the best followed by bio-inspired routing (Bio). This is because WRP follows the optimal path from source to sink (Figure 10.1(a)); in terms of the average delay in packet transfer in terms of the number of hops, Bio greatly outperforms other strategies in all three failure conditions (Figure 10.1(b)). Our initial experiments motivate the design of *novel dynamic routing strategies* where it is possible to design a unified routing strategy which is a weighted combination of WRP, E-TORA and Bio, in which the weights can be tweaked to meet changing requirements of the network, as in software defined networking. For instance, if a low data delivery delay is preferred, high role A nodes can be preferred as next hops; conversely, the weight corresponding to WRP can be increased to meet high data delivery needs.

10.2. HUB AND SPOKE ARCHITECTURE

The three tier characterization (see Section 3) can be utilized to identify significant patterns in the organization among high node motif central (NMC) nodes (defined as nodes with $\delta > 100$, as per Eq. 4.1) in tier 2. Figure 10.2 shows a schematic representation, where a few high NMC nodes in tier 2 (marked in blue) form cliques among themselves while the other high NMC nodes (shown in green) are connected to some (but not all) of the blue nodes. Note that both green and blue nodes are connected through bidirectional edges leading to full duplex data flow. There exists a third type of node (shown in yellow) that serve as intermediaries for information flow between the blue nodes.

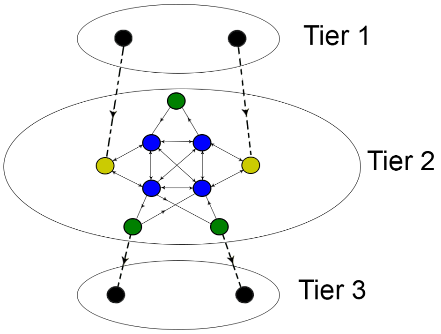


Figure 10.2. Hub and spoke architecture. Schematic representation of organization of motif central nodes within tier 2 of a TRN.

This arrangement among the high NMC tier 2 nodes is similar to a hub-and-spoke architecture with majority of the tier 1 and 3 nodes being directly connected to high NMC nodes in tier 2 [165]. We intuit that in such an architecture role A motif central nodes play the role of information spreaders, while the green nodes provide fault tolerance against failures of the blue node failures; our investigation reveals that the yellow nodes, possessing high role B motif centrality, provide edge level fault tolerance by activating the indirect path of the FFL when the direct path is congested or error prone. We believe that the organization of high NMC nodes in TRN can explain the robustness of TRNs and further motivate the design of fault-tolerant communication network topologies.

10.3. BALANCED AND UNBALANCED TRIADS

In Section 2.1 we discussed coherent and incoherent FFL motifs based on the signs on directed edges that can cause acceleration or delay in information flow in TRN. It is worth noting that this coherence or incoherence of FFLs resembles the idea of balanced and unbalanced triads in social networks. In a signed network where positive and negative edges signify friendship and enmity between a node pair, three positive edges in a balanced triad works on the following principles: *the friend of my friend is my friend*, *the friend of my enemy is my enemy*, *the enemy of my friend is my enemy* and *the enemy of my enemy is my friend*. Leskovec et al. studied the signed interaction among the entities in social network to develop a theory that explains observed edge signs and the underlying social mechanisms [168]. We intuit that the use of influence diffusion mechanisms [161, 169] in the elaborate analysis of the FFLs can help identify influencial FFL motifs that can amplify or dampen spread of information in signed social and biological network topologies.

10.4. DESIGN OF SMART TOPOLOGIES

Abdelzaher et al. showed that TRNs are naturally optimized for average shortest path (Eq. 3.2) between TFs and genes [170]. In other words, TRNs will exhibit a lower average shortest path compared to a randomized network with similar in- and out-degree distributions. We believe that defining average shortest path in terms of FFL motif centrality can yield new insights into the topological robustness of TRNs. This is because, not only are FFL motifs closely linked to TRN robustness (as shown in Section 3.2.7), but Gorochowski et al. showed that *motif clustering* (defined in Appendix 1 Section 4) leads to specific functional properties in biological networks and nodes with high FFL *motif clustering diversity* (MCD) (defined in Section 4.1) are amongst the most functionally significant TFs/genes in TRNs in terms of information spread [117]. Thus, it is worth exploring whether TRNs achieve efficient information dissemination by employing the high FFL motif central edges for data flow. One way to validate this hypothesis is to re-apply the optimization problem explained by Abdelzaher et al. [170] on several weighted versions of TRN subgraphs, where the weight on each edge equals its reciprocal FFL edge motif centrality. If TRNs still exhibit a lower δ than its randomized counterpart, it stands to reason that smart network topologies can be realized by pushing the bulk of the data packets into high FFL motif central nodes and links.

APPENDIX A.

FEED FORWARD LOOP (FFL) MOTIFS

1. ABUNDANCE OF FFL VS. FBL

Table 1 shows the comparison in the abundance of FFL vs. FBLs in *E. coli*, *S.cerevisiae*, human and mouse TRNs.

Table 1. Abundance of FFL and FBL motifs in TRN.

TRN type	FFL	FBL
E. coli	4798	12
S. Cerevisiae	4115	39
Human	7557	789
Mouse	4328	492

2. FFL AS BUILDING BLOCKS OF LARGER MOTIFS

FFLs are building blocks to larger TRN motifs, as shown in case of *E. coli* TRN in Figure 1.

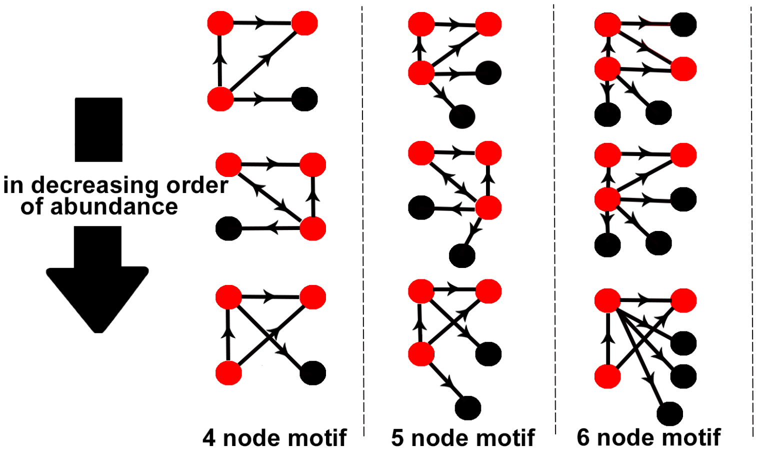


Figure 1. Abundant 4,5 and 6 node motifs in *E. coli* TRN contain FFL motifs.

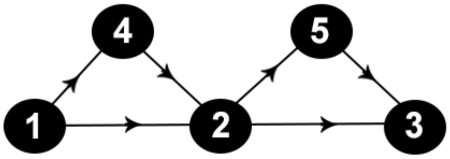


Figure 2. Path p consisting of several FFL motifs.

3. MOTIF PATH ENUMERATION

In this section, we propose a simple heuristic to analyze whether the direct and indirect links of FFL motifs are responsible for creating a majority of the paths in TRNs.

Let us consider a path $p = \{u_1, u_2, \cdots, u_n\}$. The path may contain of several FFL motifs with $e(u_i, u_{i+1})$ as direct links ($\forall i = 1, 2, \cdots (n-1)$). We know that $\phi_d(e(u_i, u_{i+1})$ is the number of nodes v such that $e(u_i, u_{i+1})$ is a direct link in the FFL motif (u_i, v, u_{i+1}).

For instance, in Figure 2, we have $p = \{1, 2, 3\}$, where edges $e(1, 2)$ and $e(2, 3)$ are direct links to FFL motifs $(1, 4, 2)$, $(2, 5, 3)$. Thus, $\phi_d(e(1, 2)) = \{4\}$ and $\phi_d(e(2, 3)) = \{5\}$.

We next apply an enumerative strategy to determine the number of paths created by FFL motifs: given a path p, replace each direct link (u_i, u_{i+1}) by an indirect path $\{u_i, v, u_{i+1}\}$ to get a new path p' created by FFL motifs (Algorithm Enumerative Approach).

Algorithm 7 Enumerative approach

```
procedure
    $i = 0$
    for $e(u_i, u_{i+1})$ do
        $p' = \{u_1, u_2, \cdots, u_i, v, u_{i+1}, \cdots, u_n\}\ \forall v \in \phi_d(e(u_i, u_{i+1}))$
```

Using this enumerative approach, for $p =< 1, 2, 3 >$, we obtain new paths $p' = \{1, 4, 2, 3\}, \{1, 2, 5, 3\}, \{1, 4, 2, 5, 3\}$.

In path $p = \{u_1, u_2, \cdots, u_n\}$, number of simple paths between u_1 and u_n formed due to FFL motifs is given by:

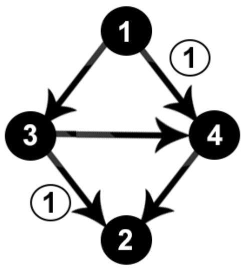


Figure 3. Example subgraph G with edge ϕ_d values shown in circles.

$$\prod_{i=1}^{n} |\phi_d(e(u_i, u_{i+1}))| + 1 \tag{1}$$

Going back to Figure 2, since $|\phi_d(1, 2)| = 1$ and $|\phi_d(2, 3)| = 1$, number of simple paths between 1 and 3 = $(|\phi_d(1, 2)| + 1) \times (|\phi_d(2, 3)| + 1) = 4$.

Redundancy in path enumeration: Let us now consider a subgraph G (Figure 3). There are a total of three paths between nodes 1 and 2, i.e., $p_1 = \{1, 3, 2\}$, $p_2 = \{1, 4, 2\}$ and $p_3 = \{1, 3, 4, 2\}$.

Although there are 3 paths between nodes 1 and 2, our enumeration equation (Eq. 1) returns 4 paths. This is because path $p_3 = \{1, 3, 4, 2\}$ is common to both direct links $e(1, 4)$ and $e(3, 2)$ and is counted twice during path enumeration.

We infer that Eq. 1 may have redundancy in path enumeration if two motifs share an indirect link. Based on these observations, *we propose a simple heuristic to determine the fraction of simple paths created as a result of the direct as well as indirect path of FFL motifs.* Here we present a simple illustration of the algorithm for subgraph G (shown in Figure 3).

Algorithm 8 Path enumeration

```
procedure
        Graph $G$, $pLimit\ score = \frac{C}{T}$
    // Calculate $\phi(e)$ for all $e \in E(G)$
    // Total paths count
    $T = 0$
    // Total paths count created by FFLs
    $C = 0$
    for $u \in V(G)$ do
        for $v \in V(G)$ do
            $P$ : List of simple paths between $u$ and $v$ of length $\leq pLimit$
            $T = T + |P|$
            for $p \in P$ do
                Let $p =< u, u_2, u_3, \cdots, v >$
                $P'$ : All possible paths between $u$ and $v$ generated using $\phi$ values using enumerate approach of length $\leq pLimit$
                $C = C + |P'|$
                $P = P - P'$
```

Illustrative example: Here for each pair of nodes $u, v \in V$, we consider a **score** as the ratio between the number of paths produced as a result of the direct and indirect path of FFL motifs to the total number of paths.

Paths between node pair $(1, 2)$*:* There are 3 paths namely, $1 \rightarrow 3 \rightarrow 2$, $1 \rightarrow 4 \rightarrow 2$ and $1 \rightarrow 3 \rightarrow 4 \rightarrow 2$ all of which are generated as a result of direct and indirect path of FFLs $(1, 3, 4)$ and $(3, 4, 2)$. **Score:** $\frac{3}{3}$.

Paths between nodes $(1,3)$ *and* $(1,4)$*:* There is 1 paths namely, $1 \rightarrow 3$ which is not using the direct and indirect path of FFL $(1,3,4)$. Similarly, there are two paths between $1 \rightarrow 4$: $1 \rightarrow 4$, $1 \rightarrow 3 \rightarrow 4$ that use the FFL $(1,3,4)$. **Scores:** $\frac{0}{1}$ and $\frac{2}{2}$.

Paths from node 2*:* Node 2 does not have any outgoing edges. Scores for node pairs $(2,1)$, $(2,3)$ and $(2,4)$ are all $\frac{0}{0}$.

Similarly, scores for paths between node pairs $(3,1)$, $(3,2)$, $(3,4)$ are $\frac{0}{0}$, $\frac{2}{2}$, $\frac{0}{1}$; and for $(4,1)$, $(4,2)$ and $(4,3)$ $\frac{0}{0}$, $\frac{0}{1}$ and $\frac{0}{0}$.

Total score of $(\frac{3+0+2+2+0+0}{3+1+2+2+1+1} =)0.7$ implies that 70% of the total paths between all pairs of nodes utilize the direct and indirect paths of FFL motifs $(1,3,4)$ and $(3,4,2)$ in subgraph G.

Here we propose a simple heuristic (Algorithm Path Enumeration) to determine the ratio between total number of simple paths created by FFL motifs to the total number of paths between all pair of nodes, while discounting the redundancy in path enumeration.

Algorithm description: It takes two input parameters: directed input graph G and maximum considered path-length $pLimit$. For every pair of nodes u and v such that v is reachable from u, it uses the Python Networkx library [171] to determine the list of simple paths P of length $\leq pLimit$. For every path p in list P, the algorithm applies the enumerative approach to list all simple paths P' created from path p via FFL motifs. The heuristic handles redundancy in path enumeration by removing all paths in P' from P. Finally, the algorithm returns the **score** as the ratio between total paths created by FFL motifs, C, and total number of simple paths (of length less than or equal to $pLimit$), T.

4. MOTIF BASED METRICS

Below are the details of the metrics used to validate the functional properties of motif central nodes in TRN.

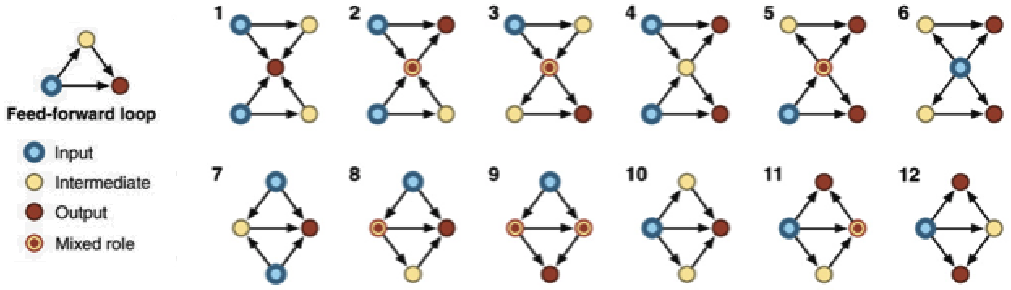


Figure 4. 12 types in Motif Clustering Diversity (MCD) (taken from [117]).

4.1. MOTIF CLUSTERING DIVERSITY

For any node, one can extract all FFL motifs that contain the selected node as a member. For all pairs of these motifs it is possible to create 12 possible configurations (shown in Figure 4). *Motif clustering diversity (MCD) is defined as the number of different motif clustering types (i.e., configurations) that a node takes part in.* Its value ranges between 0 and 12. Gorochowski et al. showed that several high MCD nodes in a TRN act as global regulators [117].

4.2. BIOLOGICAL PATHWAYS

A biological pathway is a series of actions among molecules in a cell that enable interaction among genes, molecules and cells. Such a pathway can trigger the assembly of new molecules, such as a fat or protein [172, 173]. Pathways can also turn genes on and off, or make a cell move. There are different types of biological pathways that help in signal transduction pathways, gene regulation and metabolism.

4.3. K-SHELL DECOMPOSITION

k-shell decomposition is the process of pruning of nodes with the lowest degree in an undirected graph. Each step of pruning is denoted by the variable $k = 1, 2, 3 \cdots$. After every step of pruning, the resultant **k-core subgraph** is the maximal subgraph such that every vertex has degree at least k. The integer value k is attached to the set of vertices that are part of the k^{th}core but not part of the $(k+1)^{th}$ core. Evidently, small k-values correspond to nodes in the periphery of the network and the innermost network core corresponds to large k-values.

5. FUNCTIONAL ROLE OF MOTIF CENTRAL NODES

Here we report the top 10 TFs and genes with high role A and B properties (but not featuring in top 10 high degree nodes) in Escherichia coli (*E. coli*), *S. Cerevisiae* and Mus musculus (mouse) TRN that play a role in fault tolerance. Additionally, we also report the MCD and k-values of the respective TFs. It is noteworthy that most of the chosen nodes exhibit high MCD and k-values showing their properties as global regulators and information spreaders in the networks. Also, the highest k-value for *E. coli*, *S. Cerevisiae* TRNs and mouse are 8 and 11, respectively.

5.0.1. Mouse TRN:. The functional properties of motif central nodes in the Mouse TRN are reported in Table 4. Each of the top nodes exhibit appreciable role A and role B type motif centralities. Although the signalling pathway count for some of these nodes are quite low, they all demonstrate high MCD values to underline their role as global regulators and information spreaders in the network. This can be because signalling pathway participation of all these nodes have not yet been confirmed experimentally in KEGG. Of particular note is the Crebbp gene having slightly lower MCD (of 8), but it actually does participate in an appreciable number of signalling pathways. Since each of these nodes demonstrate appreciable role B centralities, we showcase their involvement in fault tolerance related pathways from published literature in the following; only five genes/TFs are reported in this validation study while the fault tolerance of the other reported nodes have not been confirmed yet in the literature. With regard to the $k-$shell value, all the nodes discussed in Table 4 exhibit the highest $k-$shell value observed in mouse TRN (equal to 8), suggesting that they are located in the network core and play a role in information spread in the network.

Table 2. Motif centrality, MCD, pathways and k-values for Mouse TRN.

	Roles				
TF/Gene	**A**	**B**	**MCD**	**Pathway**	**k-value**
Myc	70	139	12	54	8
Pou5f1	50	82	12	1	8
Nfe2l2	85	76	11	4	8
Crebbp	90	49	8	26	8
Sox2	51	62	11	2	8
Snai1	49	61	11	1	8
Ctnnb1	52	46	12	26	8
Myod1	38	45	12	0	8
Sp7	20	47	11	21	8
Cebpb	43	39	12	4	8

- The MYC proto-oncogene is a gene product that coordinates the transcriptional regulation of a multitude of genes that are essential to cellular programs required for normal as well as neoplastic cellular growth and proliferation, including cell cycle, self-renewal, survival, cell growth, metabolism, protein and ribosomal biogenesis, and differentiation [174]. This demonstrates both its role A and role B properties.

- Nfe2l2 is a transcription activator that binds to antioxidant response (ARE) elements in the promoter regions of target genes and is important for the coordinated upregulation of genes in response to oxidative stress and the regulation of cellular redox conditions. It may also be involved in the transcriptional activation of genes of the beta-globin cluster by mediating enhancer activity of hypersensitive site 2 of the beta-globin locus control region. Hence, this TF plays a major role in fault tolerance.

- Snai1 expression has been linked to enhanced cellular survival in mouse embryos and cell lines (Vega et al, 2004) and in the context of chemoresistance in tumours where Snai1 expression can enhance resistance of cells to stress-induced apoptosis (Lim et al, 2013). Snai1 expression in keratinocytes can also enhance survival following stress (De Craene et al, 2014) [175] thereby demonstrating its role B properties.

- When mouse myoblasts or satellite cells differentiate in culture, the expression of myogenic regulatory factor, $MyoD$, is downregulated in a subset of cells that do not differentiate. The mechanism involved in the repression of $MyoD$ expression remains largely unknown. A stress-response pathway repressing $MyoD$ transcription was reported to be transiently activated in mouse-derived C2C12 myoblasts growing under differentiation-promoting conditions [176] highlighting its role B property.

- SOX proteins are involved in multiple events, from maintenance of stem cells pluripotency, to driving their terminal differentiation into specialized cell types. The $SOX2$ transcription factor is pivotal for early development and the maintenance of undifferentiated embryonic stem cells (ESCs). This transcription factor plays a critical role in directing the differentiation to neural progenitors and in maintaining the properties of neural progenitor stem cells [177] thereby demonstrating both role A and role B properties.

5.1. *E. COLI* TRN

- This regulation occurs even when the iron-binding site of Fur is compromised leading to the hypothesis that Fur senses iron and pH separately. Mutations in fur render the cell acid sensitive, but which component of acid stress (H+ or weak acid concentration) is countered by the Fur-regulated ASPs is not known [178].

- Acid resistance (AR) is perceived to be an important property of Escherichia coli, enabling the organism to survive gastric acidity and volatile fatty acids produced as a result of fermentation in the intestine. The ability to resist these acid stresses is believed to be necessary for this organism to colonize and establish a commensal relationship with mammalian hosts. In addition, the low infectious dose associated with enterohemorrhagic *E. coli* serotype O157:H7 is attributed to its acid-resistant nature [179].

Table 3. Motif centrality, MCD, pathways and k-values for *E. coli* TRN.

	Roles				
TF/Gene	**A**	**B**	**MCD**	**Pathway**	**k-value**
fur	88	133	8	26	6
gadX	88	133	12	0	6
gadE	63	91	11	0	6
gadW	39	50	7	0	6
marA	50	77	9	19	6
cpxR	71	56	2	2	5
soxS	61	36	6	0	6
fhlA	42	93	4	0	5
glnG	45	52	5	1	4
lexA	55	55	0	0	2

- $marA$ reported to play a role in adaptive response. This review will focus on $MarA$, $SoxS$ and Rob of Escherichia coli. These homologous regulators are excellent examples of global regulators that are part of multiple regulatory mechanisms necessary for the adaptive response. $MarA$, $SoxS$ and Rob, which are all members of the $AraC$ family of proteins, respond to many stimuli, including changing pH, the presence of antibiotics, oxidative stressors and organic solvents, all of which threaten survival [180].

- $CpxA$ and $CpxR$ were proposed to regulate an envelope stress response that monitored and mediated adaptation to misfolded, secreted proteins [181].

- $SoxS$ protein (M(r) of only 12,900) is a direct transcriptional activator of the oxidative stress genes of the $soxRS$ regulon, although the possible involvement of other proteins in transcription activation by $SoxS$ has not been ruled out [182].

- $Hyd-3$ or $FhlA$ itself as well as $Hyd-4$ were osmosensitive or play a role in osmoregulation. It could not be ruled out that $Hyd-4$ has a crucial role in osmotic stress response, which could result in interaction with other proteins including F0F1 to stabilize the cell turgor and maintain internal pH and Δp [183].

Table 4. Motif centrality, MCD, pathways and k-values for *S. Cerevisiae* TRN.

	Roles				
TF/Gene	**A**	**B**	**MCD**	**Pathway**	**k-value**
yap6	362	369	12	0	11
Rox1	213	193	12	0	11
PHD1	176	188	12	0	11
SWI4	120	242	9	3	11
NRG1	315	46	7	0	11
MSN4	59	203	12	4	11
MSN2	159	68	9	0	11
XBP1	20	173	12	0	11
Yap1	72	125	12	0	11
GCN4	117	74	12	0	11

5.2. *S. CEREVISIAE* TRN

- Four top candidates, $Cin5$, $Skn7$, $Phd1$, and $Yap6$, all known to be associated with stress response gene regulation, were experimentally confirmed to physically interact with $Tup1$ and/or $Ssn6$ [184].

- In this work we have overtaken a study trying to obtain integrative information about the role of the *S. Cerevisiae* genes $IXR1$, $ROX1$ and $SKY1$ in the oxidative stress response induced by As (V), Cd (II) and cisplatin in terms of modulation of four enzymatic activities [185].

- $Msn4$ is two zinc-finger transcription factor initially described as mediators of the *S. Cerevisiae* general stress response because of their capacity to jointly modulate the expression of a large battery of unrelated genes in response to a shift to suboptimal growth conditions [186].

- The *XBP*1 promoter contains several stress-regulated elements, and its expression is induced by heat shock, high osmolarity, oxidative stress, DNA damage, and glucose starvation [187].

- *Yap*1 (21) controls a large oxidative stress response regulon of at least 32 proteins [188].

- *Gcn*4 Is Required for the Response to Peroxide Stress in *S. Cerevisiae* [189].

APPENDIX B.

EDGE REWIRING AND ROBUST WIRELESS SENSOR NETWORK

1. CORE PERIPHERY PHENOMENON

Core nodes are high degree nodes that are densely connected with each other. Analogously, nodes having very few links with higher ranked nodes are considered to be a members of the periphery. Such core-periphery property is found in social networks [190] and internet [191].

One measure of core-periphery phenomenon in networks is the rich club coefficient introduced by Zhou and Mondragon [192, 193]. They defined the topological *rich club coefficient* as the proportion of edges connecting the rich core nodes with respect to all possible number of edges between them. Given a degree k in a network where $N_{>k}$ refers to the nodes having a degree higher than k, and $E_{>k}$ denotes the number of edges among the $N_{>k}$ nodes in the rich club, rich club coefficient is calculated as:

$$\phi(k) = \frac{2 \times E_{>k}}{N_{>k} \times (N_{>k} - 1)} \tag{1}$$

Evidently if $\phi(k) = 0$ the well-connected nodes do not share any links, if $\phi(k) = 1$ the rich-nodes form a clique.

It is known that protein-protein networks, consisting of proteins as nodes and their interaction as edges lack rich cores because the well-connected nodes tend to reside in different communities and do not form a direct rich club with each other [194]. Let us now evaluate the rich club coefficient of TRNs.

Figure 1 shows the degree vs. rich club coefficient ϕ for TRNs. Evidently, the rich-club coefficient does not exceed 0.4 and high degree central nodes in TRN do not form tightly interconnected communities. Thus, both *E. coli* and Yeast TRNs lack the rich club phenomenon.

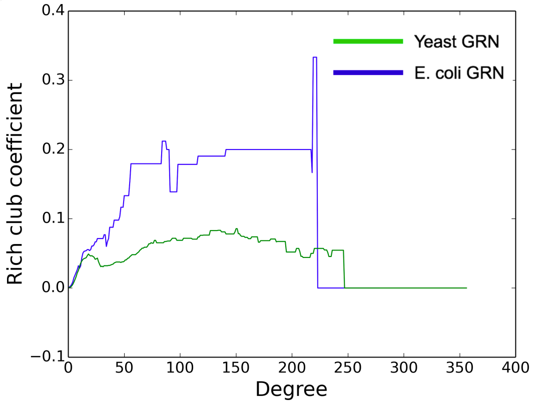


Figure 1. Degree vs. rich club coefficient for *E. coli* and Yeast TRNs.

2. FFLS ACROSS TIERS

Table 1 summarizes the percentage of Feed Forward Loops (FFLs) across tiers.

Table 1. Percentage of FFLs present across tiers.

Tiers	*1-2*	*1-3*	*2-2*	*2-3*	*1-2-3*
E. coli	0.08%	0.0%	11.4%	86.2 %	2.2%
Yeast	0.09%	0.0%	4.9%	83.2 %	10.8%

Table 1 shows that the maximum number of FFL motifs are present between tiers 2 and 3.

3. MAPPING ALGORITHM

Here is the mapping algorithm between any input topology $G_g(V_g, E_g)$ and already deployed WSN topology $G_w(V_w, E_w)$. Each node $u \in V_w$ is deployed in position C_u.

Algorithm description: In Algorithm 1, V_g and V_w are ranked in the non-increasing order of pagerank [195] (Line 1). Higher the pagerank of a node, greater is its reachability. Thus, the intuition behind the use of pagerank in this algorithm is to find a mapping between any $u \in V_w$ and $v \in V_g$ which have similar reachability from other nodes. In course of this algorithm, we define the mapping function $m : V_w \rightarrow V_g$. The highest ranked $v \in V_w$ is

Algorithm 9 Graph mapping algorithm

```
$G_g(V_g, E_g)$, $G_w(V_w, E_w)$, $C$ **m**
Nodes $V_w$ and genes $V_g$ are ranked in non-increasing order of pagerank
$\mathbf{m} = \emptyset \quad v = V_w[0]$
for $u \in V_g$ do
    if $\delta(u) \leq \delta(v)$ then $\mathbf{m}[v] = u \quad V_g = V_g - u \quad V_g = V_g - u$   break
    for $u \in V_g$ do
        for $v \in V_w$ do $flag = False$
            for $w \in \mathbf{m}$ do
                if $e(u, \mathbf{m}[w]) \in E_g$ and $e(v, w) \notin E_w$ then $flag = True$   break
                if flag = False then $\mathbf{m}[r] = g \quad V_w = V_w - v$
```

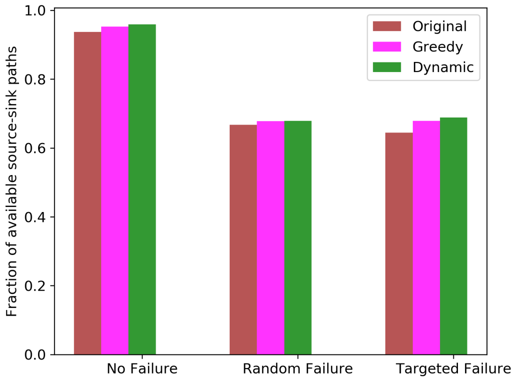


Figure 2. Performance of WSN under failure conditions. Fraction of source nodes connected to at least one sink nodes under conditions of (a) No node failure (b) Random failure and (c) Targeted failure of 20% nodes.

mapped to the highest ranked $u \in V_g$ if $\delta(u) \leq \delta(v)$, where $\delta(u)$ denotes the degree of a node u (Lines 4 - 6). Subsequently, each unmapped node $v \in V_w$ is mapped to $u \in V_g$, if, for each node $w \in V_w$ mapped to corresponding node $m(w) \in V_g$ and there exists $e(u, m(w)) \in E_g$, the corresponding edges $e(v, w) \in E_w$ must exist in E_w. The mapping function m is returned. Finally, each node u in mapped-WSN G'_w is deployed in position C_u.

4. WSN PERFORMANCE

The WSN performance of any topology depends on the availability of communication pathway between the source nodes and at least one sink. TRNs tend to preserve comparable number of source to sink communication pathways under conditions of no node failure, random node failure and targeted node failure of up to 20% nodes. To illustrate our point, we estimate the average fraction of source nodes connected to at least one sink nodes in the three TRN topologies for 50 topologies of 300 nodes.

Figure 2 shows that a large fraction of source nodes connected to at least 1 sink for dynamic TRN is comparable to that of original and greedily rewired topologies. Consequently, the performance of rewired TRN-based WSNs exhibit a $5 - 10\%$ improvement in PDR and network latency over its original counterpart.

APPENDIX C.

PUBLICATIONS

*co-primary author

1. PEER-REVIEWED CONFERENCE PAPERS

- S. Roy, N. Ghosh, P. Ghosh and S.K. Das. "bioMCS: A Bio-inspired Collaborative Data Transfer Framework over Fog Computing Platforms in Mobile Crowdsensing" International Conference on Distributed Computing and Networking (ICDCN) 2019 (accepted).

- V.K.Shah*, S. Roy*, S. Silvestri, and S. K. Das. "Bio-DRN: Bio-inspired Disaster Response Network" under review at 16th IEEE International Conference on Mobile Ad-Hoc and Smart Systems (MASS) (accepted).

- S. Roy*, N. Ghosh*, S.K. Das. "bioSmartSense: A Bio-inspired Data Collection Framework for Energy-efficient, QoI-aware Smart City Applications" In IEEE Pervasive Computing and Communications 2019.

- S. Roy, M. Raj, P. Ghosh, S.K. Das. "Role of motifs in topological robustness of gene regulatory networks." In 2017 IEEE International Conference on Communications (ICC), pp. 1-6.

- S. Roy, V.K.Shah, S. K. Das. "Characterization of E. coli Gene Regulatory Network and its Topological Enhancement by Edge Rewiring." In 9th EAI International Conference on Bio-inspired Information and Communications Technologies (BICT) 2015, pp. 391-398. 2015.

2. JOURNAL PAPERS

- S. Roy, V.K.Shah, S. K. Das. "Design of Robust and Efficient Topology using Enhanced Gene Regulatory Networks." IEEE Transactions on Molecular, Biological and Multi-Scale Communications (2019).

3. SHORT AND POSTER PAPERS

- V.K.Shah*, S. Roy*, S. Silvestri, and S. K. Das. "Towards energy-efficient and robust disaster response networks." In Proceedings of the 20th International Conference on Distributed Computing and Networking, pp. 397-400. ACM, 2019.

- S. Roy and S. K. Das. "A Bio-inspired Approach to Design Robust and Energy-efficient Communication Network Topologies" PhD Forum on IEEE Pervasive Computing and Communications (PerCom) 2019.

4. PAPERS IN PREPARATION/UNDER REVIEW

- S. Roy, P. Ghosh, D. Barua and S.K. Das. "Motifs enable communication efficiency and fault-tolerance in transcriptional networks" under review at Nature Scientific Reports.

- S. Roy, N. Ghosh and S.K. Das. "bioSmartSense+: A Bio-inspired Probabilistic Data Collection Framework for Priority-based Event Reporting in IoT Environments" under review at Pervasive and Mobile Computing.

- S. Roy, N. Ghosh, P. Ghosh and S.K. Das. "Structure and Topology of Transcriptional Regulatory Networks and their Applications in Bio-inspired Networking: A survey" to be submitted as ACM Computing Surveys.

- S. Roy, P. Ghosh and S.K. Das. "A computational framework to identify minimal druggable targets in transcriptional networks" to be submitted at Nature Physics.

REFERENCES

[1] F. Dressler and O. B. Akan. A survey on bio-inspired networking. *Computer Networks*, 54(6):881–900, 2010.

[2] A. Fukushima, S. Kanaya, and K. Nishida. Integrated network analysis and effective tools in plant systems biology. *Frontiers in plant science*, 5:598, 2014.

[3] U. Aickelin, D. Dasgupta, and F. Gu. Artificial immune systems. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, Editors: Burke, Edmund K., Kendall, Graham (Eds.)(2014)*, 2014.

[4] N. Forbes. Biologically inspired computing. *Computing in Science & Engineering*, 2(6):83–87, 2000.

[5] C Kotteeswaran and A Rajesh. A survey of diverse nature bio-inspired computing models. In *Second International Conference on Current Trends In Engineering and Technology-ICCTET 2014*, pages 120–124. IEEE, 2014.

[6] M. Meisel, V. Pappas, and L. Zhang. A taxonomy of biologically inspired research in computer networking. *Computer Networks*, 54(6):901–916, 2010.

[7] V. Sevim and P. A. Rikvold. Chaotic gene regulatory networks can be robust against mutations and noise. *Journal of theoretical biology*, 253(2):323–332, 2008.

[8] N. Noman, T. Monjo, P. Moscato, and H. Iba. Evolving robust gene regulatory networks. *PloS one*, 10(1):e0116258, 2015.

[9] Y. Fu, L. R. Jarboe, and J. A. Dickerson. Reconstructing genome-wide regulatory network of e. coli using transcriptome data and predicted transcription factor activities. *BMC bioinformatics*, 12(1):233, 2011.

[10] T. R. Sorrells and A. D. Johnson. Making sense of transcription networks. *Cell*, 161(4):714–723, 2015.

[11] T. Schaffter, D. Marbach, and D. Floreano. Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, 27(16):2263–2270, 2011.

[12] I. Pournara and L. Wernisch. Factor analysis for gene regulatory networks and transcription factor activity profiles. *BMC bioinformatics*, 8(1):61, 2007.

[13] H. Han, H. Shim, D. Shin, J. E. Shim, Y. Ko, J. Shin, H. Kim, A. Cho, E. Kim, T. Lee, et al. Trrust: a reference database of human transcriptional regulatory interactions. *Scientific reports*, 5:11432, 2015.

[14] H. Han, J. Cho, S. Lee, A. Yun, H. Kim, D. Bae, S. Yang, C. Y. Kim, M. Lee, E. Kim, et al. Trrust v2: an expanded reference database of human and mouse transcriptional regulatory interactions. *Nucleic acids research*, 46(D1):D380–D386, 2017.

[15] E. Wong, B. Baur, S. Quader, and C. Huang. Biological network motif detection: principles and practice. *Briefings in bioinformatics*, 13(2):202–215, 2011.

[16] U. Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450–461, 2007.

[17] T. I. Lee, N. J Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, et al. Transcriptional regulatory networks in saccharomyces cerevisiae. *science*, 298(5594):799–804, 2002.

[18] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature genetics*, 31(1):64, 2002.

[19] S. Mangan and U. Alon. Structure and function of the feed-forward loop network motif. *Proceedings of the National Academy of Sciences*, 100(21):11980–11985, 2003.

[20] E. Estrada. *The structure of complex networks: theory and applications*. Oxford University Press, 2012.

[21] R. J. Prill, P. A. Iglesias, and A. Levchenko. Dynamic properties of network motifs contribute to biological network organization. *PLoS biology*, 3(11):e343, 2005.

[22] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics*, 20(11):1746–1758, 2004.

[23] F. Schreiber and H. Schwöbbermeyer. Mavisto: a tool for the exploration of network motifs. *Bioinformatics*, 21(17):3572–3574, 2005.

[24] S. Wernicke and F. Rasche. Fanmod: a tool for fast network motif detection. *Bioinformatics*, 22(9):1152–1153, 2006.

[25] B. D. McKay et al. *Practical graph isomorphism*.

[26] M. Zuba. A comparative study of network motif detection tools.

[27] M. E. Wall, M. J. Dunlop, and W. S. Hlavacek. Multiple functions of a feed-forward-loop gene circuit. *Journal of molecular biology*, 349(3):501–514, 2005.

[28] E. Alm and A. P. Arkin. Biological networks. *Current opinion in structural biology*, 13(2):193–202, 2003.

[29] H. Ma, J. Buer, and A. Zeng. Hierarchical structure and modules in the escherichia coli transcriptional regulatory network revealed by a new top-down approach. *BMC bioinformatics*, 5(1):199, 2004.

[30] M Madan Babu, Nicholas M Luscombe, L Aravind, Mark Gerstein, and Sarah A Teichmann. Structure and evolution of transcriptional regulatory networks. *Current opinion in structural biology*, 14(3):283–291, 2004.

[31] O. Shoval and U. Alon. Snapshot: network motifs. *Cell*, 143(2):326–326, 2010.

[32] A. Martínez-Antonio, S. C. Janga, and D. Thieffry. Functional organisation of escherichia coli transcriptional regulatory network. *Journal of molecular biology*, 381(1):238–247, 2008.

[33] R. Pinho, V. Garcia, M. Irimia, and M. W. Feldman. Stability depends on positive autoregulation in boolean gene regulatory networks. *PLoS computational biology*, 10(11):e1003916, 2014.

[34] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella. Energy conservation in wireless sensor networks: A survey. *Ad hoc networks*, 7(3):537–568, 2009.

[35] B. K. Kamapantula, A. F. Abdelzaher, M. Mayo, E. J. Perkins, S. K. Das, and P. Ghosh. Quantifying robustness in biological networks using ns-2. In *Modeling, Methodologies and Tools for Molecular and Nano-scale Communications*, pages 273–290. Springer, 2017.

[36] H. De Jong. Modeling and simulation of genetic regulatory systems: a literature review. *Journal of computational biology*, 9(1):67–103, 2002.

[37] D. S. Banks and C. Fradin. Anomalous diffusion of proteins due to molecular crowding. *Biophysical journal*, 89(5):2960–2971, 2005.

[38] T. Schlitt and A. Brazma. Current approaches to gene regulatory network modelling. *BMC bioinformatics*, 8(6):S9, 2007.

[39] P. Ghosh, M. Mayo, V. Chaitankar, T. Habib, E. Perkins, and S. K. Das. 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*, pages 160–165. IEEE, 2011.

[40] B. K. Kamapantula, A. Abdelzaher, P. Ghosh, M. Mayo, E. Perkins, and S. K. Das. Performance of wireless sensor topologies inspired by e. coli genetic networks. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, pages 302–307. IEEE, 2012.

[41] B. K. Kamapantula, A. Abdelzaher, P. Ghosh, M. Mayo, E. J. Perkins, and S. K. Das. Leveraging the robustness of genetic networks: a case study on bio-inspired wireless sensor network topologies. *Journal of Ambient Intelligence and Humanized Computing*, 5(3):323–339, 2014.

[42] A. Laszka, L. Buttyán, and D. Szeszlér. Designing robust network topologies for wireless sensor networks in adversarial environments. *Pervasive and Mobile Computing*, 9(4):546–563, 2013.

[43] A. Nazi, M. Raj, M. Di Francesco, P. Ghosh, and S. K. Das. Deployment of robust wireless sensor networks using gene regulatory networks: An isomorphism-based approach. *Pervasive and Mobile Computing*, 13:246–257, 2014.

[44] A. Nazi, M. Raj, M. Di Francesco, P. Ghosh, and S. K. Das. Robust deployment of wireless sensor networks using gene regulatory networks. In *International Conference on Distributed Computing and Networking*, pages 192–207. Springer, 2013.

[45] A. Nazi, M. Raj, M. Di Francesco, P. Ghosh, and S. K. Das. Efficient communications in wireless sensor networks based on biological robustness. In *Distributed Computing in Sensor Systems (DCOSS), 2016 International Conference on*, pages 161–168. IEEE, 2016.

[46] A. Nazi, M. Raj, M. Di Francesco, P. Ghosh, and S. K. Das. Exploiting gene regulatory networks for robust wireless sensor networking. In *2015 IEEE Global Communications Conference*, pages 1–7, 2015.

[47] A. Markham and N. Trigoni. Discrete gene regulatory networks (dgrns): A novel approach to configuring sensor networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[48] H. Byun and J. Park. A gene regulatory network-inspired self-organizing control for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 11(8):789434, 2015.

[49] A. Vahdat, D. Becker, et al. Epidemic routing for partially connected ad hoc networks. *Technical Report CS-200006, Duke University*, 2000.

[50] M. Y. S. Uddin, H. Ahmadi, T. Abdelzaher, and R. Kravets. Intercontact routing for energy constrained disaster response networks. *IEEE Transactions on Mobile Computing*, 12(10):1986–1998, 2013.

[51] V. K. Shah, S. Roy, and S. K. Das. Ctr: A cluster based topological routing for disaster response networks. In *IEEE International Conference on Communications (Accepted)*, 2017.

[52] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE mobile computing and communications review*, 7(3):19–20, 2003.

[53] J. Burgess, B. Gallagher, D. Jensen, B. N. Levine, et al. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *Infocom*, 2006.

[54] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259. ACM, 2005.

[55] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In *IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW)*, pages 79–85, 2007.

[56] N. Benamar, Kamal D Singh, M. Benamar, D. El Ouadghiri, and J-M Bonnin. Routing protocols in vehicular delay tolerant networks: A comprehensive survey. *Computer Communications*, 48:141–158, 2014.

[57] K. Wei, X. Liang, and K. Xu. A survey of social-aware routing protocols in delay tolerant networks: applications, taxonomy and design-related issues. *IEEE Communications Surveys & Tutorials*, 16(1):556–578, 2014.

[58] M. Huang, S. Chen, Y. Zhu, and Y. Wang. Topology control for time-evolving and predictable delay-tolerant networks. *IEEE Transactions on Computers*, 62(11):2308–2321, 2013.

[59] Fan Li, Siyuan Chen, Minsu Huang, Zhiyuan Yin, Chao Zhang, and Yu Wang. Reliable topology design in time-evolving delay-tolerant networks with unreliable links. *IEEE Transactions on Mobile Computing*, 14(6):1301–1314, 2015.

[60] H. Chen, K. Shi, and C. Wu. Spanning tree based topology control for data collecting in predictable delay-tolerant networks. *Ad Hoc Networks*, 46:48–60, 2016.

[61] K. Jaiswal, S. Sobhanayak, BK Mohanta, and D. Jena. Iot-cloud based framework for patient's data collection in smart healthcare system using raspberry-pi. In *2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, pages 1–4. IEEE, 2017.

[62] A. Almeida, A. Fiore, L. Mainetti, R. Mulero, L. Patrono, and P. Rametta. An iot-aware architecture for collecting and managing data related to elderly behavior. *Wireless Communications and Mobile Computing*, 2017, 2017.

[63] PH Kulkarni, PD Kute, and VN More. Iot based data processing for automated industrial meter reader using raspberry pi. In *2016 International Conference on Internet of Things and Applications (IOTA)*, pages 107–111. IEEE, 2016.

[64] A. Capponi, C. Fiandrino, D. Kliazovich, P. Bouvry, and S. Giordano. A cost-effective distributed framework for data collection in cloud-based mobile crowd sensing architectures. *IEEE Transactions on Sustainable Computing*, 2(1):3–16, 2017.

[65] D Hasenfratz, O Saukh, S Sturzenegger, and L Thiele. Participatory air pollution monitoring using smartphones. *Mobile Sensing*, 1:1–5, 2012.

[66] I Schweizer, R Bärtl, A Schulz, F Probst, and M Mühläuser. Noisemap-real-time participatory noise maps. In *Second international workshop on sensing applications on mobile phones*, pages 1–5. Citeseer, 2011.

[67] K Han, C Zhang, and J Luo. Taming the uncertainty: Budget limited robust crowdsensing through online learning. *IEEE/ACM Transactions on Networking (TON)*, 24(3):1462–1475, 2016.

[68] L Wang, D Zhang, Z Yan, H Xiong, and B Xie. effsense: A novel mobile crowdsensing framework for energy-efficient and cost-effective data uploading. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(12):1549–1563, 2015.

[69] ND Lane, Y Chon, L Zhou, Y Zhang, F Li, D Kim, G Ding, F Zhao, and H Cha. Piggyback crowdsensing (pcs): energy efficient crowdsourcing of mobile sensor data by exploiting smartphone app opportunities. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 7. ACM, 2013.

[70] J Liu, L Bic, H Gong, and S Zhan. Data collection for mobile crowdsensing in the presence of selfishness. *EURASIP journal on wireless communications and networking*, 2016(1):82, 2016.

[71] PP Jayaraman, JB Gomes, H Nguyen, ZS Abdallah, S Krishnaswamy, and A Zaslavsky. Scalable energy-efficient distributed data analytics for crowdsensing applications in mobile environments. *IEEE Transactions on Computational Social Systems*, 2(3):109–123, 2015.

[72] W Sherchan, PP Jayaraman, S Krishnaswamy, A Zaslavsky, S Loke, and A Sinha. Using on-the-move mining for mobile crowdsensing. In *2012 IEEE 13th International Conference on Mobile Data Management*, pages 115–124. IEEE, 2012.

[73] C Fiandrino, F Anjomshoa, B Kantarci, D Kliazovich, P Bouvry, and JN Matthews. Sociability-driven framework for data acquisition in mobile crowdsensing over fog computing platforms for smart cities. *IEEE Transactions on Sustainable Computing*, 2(4):345–358, 2017.

[74] H Zhang, Y Wang, and CC Tan. Wd2: An improved wifi-direct group formation protocol. In *Proceedings of the 9th ACM MobiCom workshop on Challenged networks*, pages 55–60. ACM, 2014.

[75] UB Menegato, LS Cimino, S Delabida, Medeiros FA, JC Lima, and RAR Oliveira. Dynamic clustering in wifi direct technology. In *Proceedings of the 12th ACM international symposium on Mobility management and wireless access*, pages 25–29. ACM, 2014.

[76] A Laha, X Cao, W Shen, X Tian, and Y Cheng. An energy efficient routing protocol for device-to-device based multihop smartphone networks. In *2015 IEEE International conference on communications (ICC)*, pages 5448–5453. IEEE, 2015.

[77] V Arnaboldi, MGG Campana, and F Delmastro. Context-aware configuration and management of wifi direct groups for real opportunistic networks. In *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 266–274. IEEE, 2017.

[78] P Vitello, A Capponi, C Fiandrino, P Giaccone, D Kliazovich, U Sorger, and P Bouvry. Collaborative data delivery for smart city-oriented mobile crowdsensing systems. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2018.

[79] S. Roy, V.K. Shah, and S.K. Das. Characterization of e. coli gene regulatory network and its topological enhancement by edge rewiring. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pages 391–398, 2016.

[80] S. Roy, V. K. Shah, and S. K. Das. Design of robust and efficient topology using enhanced gene regulatory networks. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 2019.

[81] M. B Gerstein, A. Kundaje, M. Hariharan, S. G Landt, K. Yan, C. Cheng, X. J. Mu, E. Khurana, J. Rozowsky, R. Alexander, et al. Architecture of the human regulatory network derived from encode data. *Nature*, 489(7414):91, 2012.

[82] N. Bhardwaj, P. M. Kim, and M. B. Gerstein. Rewiring of transcriptional regulatory networks: hierarchy, rather than connectivity, better reflects the importance of regulators. *Sci. Signal.*, 3(146):ra79–ra79, 2010.

[83] A. Barabási. Scale-free networks: a decade and beyond. *science*, 325(5939):412–413, 2009.

[84] R. Albert. Scale-free networks in cell biology. *Journal of cell science*, 118(21):4947–4957, 2005.

[85] R. D. Leclerc. Survival of the sparsest: robust gene networks are parsimonious. *Molecular systems biology*, 4(1):213, 2008.

[86] Q. K. Telesford, K. E. Joyce, S. Hayasaka, J. H. Burdette, and P. J. Laurienti. The ubiquity of small-world networks. *Brain connectivity*, 1(5):367–375, 2011.

[87] L. Gu, H. L. Huang, and X. D. Zhang. The clustering coefficient and the diameter of small-world networks. *Acta Mathematica Sinica, English Series*, 29(1):199–208, 2013.

[88] L. A. N. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley. Classes of small-world networks. *Proceedings of the national academy of sciences*, 97(21):11149–11152, 2000.

[89] T. Guo. Design of genetic regulatory networks. 2014.

[90] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D-U Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4-5):175–308, 2006.

[91] M. Mayo, A. Abdelzaher, E. J. Perkins, and P. Ghosh. Motif participation by genes in e. coli transcriptional networks. *Frontiers in physiology*, 3:357, 2012.

[92] F. M. Camas and J. F Poyatos. What determines the assembly of transcriptional network motifs in escherichia coli? *PLoS One*, 3(11):e3657, 2008.

[93] A. F. Abdelzaher, A. F. Al-Musawi, P. Ghosh, M. L. Mayo, and E. J. Perkins. Transcriptional network growing models using motif-based preferential attachment. *Frontiers in bioengineering and biotechnology*, 3:157, 2015.

[94] M. Aldana, E. Balleza, S. Kauffman, and O. Resendiz. Robustness and evolvability in genetic regulatory networks. *Journal of theoretical biology*, 245(3):433–448, 2007.

[95] S. Roy, M. Raj, P. Ghosh, and S. K. Das. Role of motifs in topological robustness of gene regulatory networks. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2017.

[96] S. Mizutaka and K. Yakubo. Structural robustness of scale-free networks against overload failures. *Physical Review E*, 88(1):012803, 2013.

[97] G. Paul, T Tanizawa, S. Havlin, and H. E. Stanley. Optimization of robustness of complex networks. *The European Physical Journal B*, 38(2):187–191, 2004.

[98] C. M. Schneider, A. A Moreira, J. Andrade, S. Havlin, and H. J. Herrmann. Mitigation of malicious attacks on networks. *Proceedings of the National Academy of Sciences*, 108(10):3838–3841, 2011.

[99] H. Chan, L. Akoglu, and H. Tong. Make it or break it: Manipulating robustness in large networks. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 325–333. SIAM, 2014.

[100] A. Ma'Ayan, G. A. Cecchi, J. Wagner, A R. Rao, R. Iyengar, and G. Stolovitzky. Ordered cyclic motifs contribute to dynamic stability in biological and engineered networks. *Proceedings of the National Academy of Sciences*, 105(49):19235–19240, 2008.

[101] A. Elena, H. Ben-Amor, N. Glade, and J. Demongeot. Motifs in regulatory networks and their structural robustness. In *8th IEEE International Conference on BioInformatics and BioEngineering, 2008.*

[102] Y. Kwon and K. Cho. Quantitative analysis of robustness and fragility in biological networks based on feedback dynamics. *Bioinformatics*, 24(7):987–994, 2008.

[103] D. C. Marciano, R. C. Lua, C. Herman, and O. Lichtarge. Cooperativity of negative autoregulation confers increased mutational robustness. *Phys. Rev. Lett.*, 116:258104, Jun 2016.

[104] A. Nazi, M. Raj, M. Di Francesco, P. Ghosh, and S. K. Das. Robust deployment of wireless sensor networks using gene regulatory networks. In *ICDCN*, pages 192–207. Springer, 2013.

[105] J. Goni, A. Avena-Koenigsberger, N. V. de Mendizabal, M. P. van den Heuvel, R. F. Betzel, and O. Sporns. Exploring the morphospace of communication efficiency in complex networks. *PLoS One*, 8(3):e58070, 2013.

[106] C. R. Palmer, G. Siganos, M. Faloutsos, C. Faloutsos, and P. B. Gibbons. The connectivity and fault-tolerance of the internet topology.

[107] B. Kamapantula, A. Abdelzaher, P. Ghosh, M. Mayo, E. Perkins, and S.K. Das. Leveraging the robustness of genetic networks: a case study on bio-inspired wireless sensor network topologies. *Journal of Ambient Intelligence and Humanized Computing*, 5(3):323–339, 2014.

[108] A. Nazi, M. Raj, M. Di Francesco, P. Ghosh, and S.K. Das. Deployment of robust wireless sensor networks using gene regulatory networks: An isomorphism-based approach. *Pervasive and Mobile Computing*, 13:246–257, 2014.

[109] A. Nazi, M. Raj, M. Di Francesco, P. Ghosh, and S.K. Das. Efficient communications in wireless sensor networks based on biological robustness. *International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 161–168, 2016.

[110] U. Alon. *An introduction to systems biology: design principles of biological circuits*. CRC press, 2006.

[111] M. Newman and G. Ghoshal. Bicomponents and the robustness of networks to failure. *Physical review letters*, 100(13):138701, 2008.

[112] M. Newman. *Networks: an introduction*. Oxford university press.

[113] A. A. Hagberg, D. Schultz, and P. J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, August 2008.

[114] A. Varga and R. Hornig. An overview of the omnet++ simulation environment. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, page 60. ICST, 2008.

[115] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P Levis. Collection tree protocol. In *Proceedings of the 7th ACM conference on embedded networked sensor systems*, pages 1–14. ACM, 2009.

[116] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse. Identification of influential spreaders in complex networks. *Nature physics*, 6(11):888, 2010.

[117] T. E. Gorochowski, C. S. Grierson, and M. di Bernardo. Organization of feed-forward loop motifs reveals architectural principles in natural and engineered networks. *Science advances*, 4(3):eaap9751, 2018.

[118] M Kanehisa and Goto S. Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, 28(1):27–30, 2000.

[119] N. Guimarães-Camboa, J. Stowe, I. Aneas, N. Sakabe, P. Cattaneo, L. Henderson, M. S. Kilberg, R. S. Johnson, J. Chen, A. D. McCulloch, et al. Hif1α represses cell stress pathways to allow proliferation of hypoxic fetal cardiomyocytes. *Developmental cell*, 33(5):507–521, 2015.

[120] S. Bahari-Javan, H. Varbanov, R. Halder, E. Benito, L. Kaurani, S. Burkhardt, H. Anderson-Schmidt, I. Anghelescu, M. Budde, R. M. Stilling, et al. Hdac1 links early life stress to schizophrenia-like phenotypes. *Proceedings of the National Academy of Sciences*, page 201613842, 2017.

[121] Y. W. Yi, H. J. Kang, and I. Bae. Brca1 and oxidative stress. *Cancers*, 6(2):771–795, 2014.

[122] A. C. Dudley, D. Thomas, J. Best, and A. Jenkins. The stats in cell stress-type responses. *Cell Communication and Signaling*, 2(1):8, 2004.

[123] F. Moore, N. Naamane, M. L. Colli, T. Bouckenooghe, F. Ortis, E. N. Gurzov, M. Igoillo-Esteve, C. Mathieu, G. Bontempi, T. Thykjaer, et al. Stat1 is a master regulator of pancreatic β-cell apoptosis and islet inflammation. *Journal of Biological Chemistry*, 286(2):929–941, 2011.

[124] M. Papetti, S. N. Wontakal, T. Stopka, and A. I. Skoultchi. Gata-1 directly regulates p21 gene expression during erythroid differentiation. *Cell Cycle*, 9(10):1972–1980, 2010.

[125] F. Bonin, M. Molina, C. Malet, C. Ginestet, O. Berthier-Vergnes, M. T. Martin, and J. Lamartine. Gata3 is a master regulator of the transcriptional response to low-dose ionizing radiation in human keratinocytes. *BMC genomics*, 10(1):417, 2009.

[126] J. C. Rajapakse and P. A. Mundra. Stability of building gene regulatory networks with sparse autoregressive models. In *BMC bioinformatics*, volume 12, page S17. BioMed Central, 2011.

[127] S. Roy, M. Raj, P. Ghosh, and S. K. Das. Role of motifs in topological robustness of gene regulatory networks. In *IEEE International Conference on Communications, ICC 2017, Paris, France, May 21-25, 2017*, pages 1–6, 2017.

[128] E. Jones, T. Oliphant, and P. Peterson. {SciPy}: open source scientific tools for {Python}. 2014.

[129] A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[130] R. Noldus and P. Van Mieghem. Assortativity in complex networks. *Journal of Complex Networks*, 3(4):507–542, 2015.

[131] W. Ejaz, M. Naeem, A. Shahid, A. Anpalagan, and M. Jo. Efficient energy management for the internet of things in smart cities. *IEEE Communications Magazine*, 55(1):84–91, 2017.

[132] S. Roy, VK Shah, and SK Das. Design of robust and efficient topology using enhanced gene regulatory networks. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 2019.

[133] W. Z. Ouma, K. Pogacar, and E. Grotewold. Topological and statistical analyses of gene regulatory networks reveal unifying yet quantitatively different emergent properties. *PLoS computational biology*, 14(4):e1006098, 2018.

[134] C. H. Liu, B. Zhang, X. Su, J. Ma, W. Wang, and K. K Leung. Energy-aware participant selection for smartphone-enabled mobile crowd sensing. *IEEE Systems Journal*, 11(3):1435–1446, 2017.

[135] S. Roy, N. Ghosh, and SK Das. biosmartsense: A bio-inspired data collection framework for energy-efficient qoi-aware smart city applications.

[136] R. M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[137] S. Sakai, M. Togasaki, and K. Yamazaki. A note on greedy algorithms for the maximum weighted independent set problem. *Discrete Applied Mathematics*, 126(2-3):313–322, 2003.

[138] S. Basagni. Finding a maximal weighted independent set in wireless networks. *Telecommunication Systems*, 18(1-3):155–168, 2001.

[139] B. Gompertz. On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies. *Philosophical Transactions of the Royal Society of London*, 118.

[140] MEY Boudaren, MR Senouci, MA Senouci, and A. Mellouk. New trends in sensor coverage modeling and related techniques: A brief synthesis. In *2014 International Conference on Smart Communications in Network Technologies (SaCoNeT)*, pages 1–6. IEEE, 2014.

[141] A. Hossain, P. K. Biswas, and S. Chakrabarti. Sensing models and its impact on network coverage in wireless sensor network. In *2008 IEEE Region 10 and the Third international Conference on Industrial and Information Systems*, pages 1–5. IEEE, 2008.

[142] Y. Zou and K. Chakrabarty. Sensor deployment and target localization in distributed sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 3(1):61–91, 2004.

[143] A. Elfes. Occupancy grids: A stochastic spatial representation for active robot perception. *arXiv preprint arXiv:1304.1098*, 2013.

[144] N. Matloff. Introduction to discrete-event simulation and the simpy language. *Davis, CA. Dept of Computer Science. University of California at Davis.*

[145] RP Barnwal, N. Ghosh, SK Ghosh, and SK Das. Ps-sim: A framework for scalable simulation of participatory sensing data. In *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 195–202. IEEE, 2018.

[146] RP Barnwal, N. Ghosh, SK Ghosh, and SK Das. Ps-sim: A framework for scalable data simulation and incentivization in participatory sensing-based smart city applications. *Pervasive and Mobile Computing*, 57:64–77, 2019.

[147] [online] https://itsforkit.github.io/uploaded/surakshitLast accessed on May 14, 2018.

[148] L. Bahiense, G. Manić, B. Piva, and C. C. De Souza. The maximum common edge subgraph problem: A polyhedral investigation. *Discrete Applied Mathematics*, 160(18):2523–2541, 2012.

[149] L. A. Zager and G. C. Verghese. Graph similarity scoring and matching. *Applied mathematics letters*, 21(1):86–94, 2008.

[150] [online] https://math.mit.edu/ goemans/18433s09/matching-notes.pdf, Last accessed on May 14, 2018.

[151] A. Keränen, J. Ott, and T. Kärkkäinen. The one simulator for dtn protocol evaluation. In *International conference on simulation tools and techniques*, page 55. ICST, 2009.

[152] [online] https://wiki.openstreetmap.org/wiki/overpass_api.

[153] [online] https://github.com/julianofischer/osm2wkt.

[154] F. Ekman, A. Keränen, J. Karvo, and J. Ott. Working day movement model. In *Proceedings of the 1st ACM SIGMOBILE workshop on Mobility models*, pages 33–40. ACM, 2008.

[155] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy consumption in mobile phones: a measurement study and implications for network applications. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, pages 280–293. ACM, 2009.

[156] RR Roy. Random walk mobility. In *Handbook of Mobile Ad Hoc Networks for Mobility Models*, pages 35–63. Springer, 2011.

[157] VA Davies et al. Evaluating mobility models within an ad hoc network. Master's thesis, Citeseer, 2000.

[158] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[159] M. Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[160] J. J. Nalluri, P. Rana, D. Barh, V. Azevedo, T. N. Dinh, V. Vladimirov, and P. Ghosh. Determining causal mirnas and their signaling cascade in diseases using an influence diffusion model. *Scientific reports*, 7(1):8133, 2017.

[161] D. Kempe, J. Kleinberg, and É. Tardos. Influential nodes in a diffusion model for social networks. In *International Colloquium on Automata, Languages, and Programming*, pages 1127–1138. Springer, 2005.

[162] Maryam Hosseini-Pozveh, Kamran Zamanifar, and Ahmad Reza Naghsh-Nilchi. Assessing information diffusion models for influence maximization in signed social networks. *Expert Systems with Applications*, 119:476–490, 2019.

[163] I. Papatheodorou, N. A. Fonseca, M. Keays, Y. A. Tang, E. Barrera, W. Bazant, M. Burke, A. Füllgrabe, A. M. Fuentes, N. George, et al. Expression atlas: gene and protein expression across multiple studies and organisms. *Nucleic acids research*, 46(D1):D246–D251, 2017.

[164] Yoo-Ah Kim, Stefan Wuchty, and Teresa M Przytycka. Identifying causal genes and dysregulated pathways in complex diseases. *PLoS computational biology*, 7(3):e1001095, 2011.

[165] S Roy, P Ghosh, D Barua, and SK Das. Motifs enable communication efficiency and fault-tolerance in transcriptional networks. *Nature Scientific Reports (under review)*, 2019.

[166] S. Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mobile Networks and applications*, 1(2):183–197, 1996.

[167] F. Yu, Y. Li, F. Fang, and Q. Chen. A new tora-based energy aware routing protocol in mobile ad hoc networks. In *2007 3rd IEEE/IFIP international conference in central Asia on internet*, pages 1–4. IEEE, 2007.

[168] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1361–1370. ACM, 2010.

[169] W. Chen, W. Lu, and N. Zhang. Time-critical influence maximization in social networks with time-delayed diffusion process. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[170] Ahmed F Abdelzaher. Identifying parameters for robust network growth using attachment kernels: A case study on directed and undirected networks. 2016.

[171] A. Hagberg, P. Swart, and D. Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[172] National human genome research institute. https://www.genome.gov/about-genomics/fact-sheets/Biological-Pathways-Fact-Sheet.

[173] Kegg pathway database. http://www.genome.jp/kegg/pathway.html.

[174] S. C. Casey, V. Baylot, and D. W. Felsher. The myc oncogene is a global regulator of the immune response. *Blood*, 131(18):2007–2015, 2018.

[175] K. Horvay, T. Jardé, F. Casagranda, V. M. Perreau, K. Haigh, C. M. Nefzger, R. Akhtar, T. Gridley, G. Berx, J. J. Haigh, et al. Snai1 regulates cell lineage allocation and stem cell maintenance in the mouse intestinal epithelium. *The EMBO journal*, 34(10):1319–1335, 2015.

[176] J. Alter and E. Bengal. Stress-induced c/ebp homology protein (chop) represses myod transcription to delay myoblast differentiation. *PLoS One*, 6(12):e29498, 2011.

[177] M. Stevanović, D. Drakulić, M. Švirtlih, D. Stanisavljević, V. Vuković, M. Mojsin, and A. Klajn. Sox2 gene–master regulator of numerous cellular processes. *Biologia Serbica*, 39(1), 2017.

[178] S. Bearson, B. Bearson, and J. W. Foster. Acid stress responses in enterobacteria. *FEMS microbiology letters*, 147(2):173–180, 1997.

[179] M. Castanie-Cornet, T. A. Penfound, D. Smith, J. F. Elliott, and J. W. Foster. Control of acid resistance inescherichia coli. *Journal of bacteriology*, 181(11):3525–3535, 1999.

[180] V. Duval and I. M. Lister. Mara, soxs and rob of escherichia coli–global regulators of multidrug resistance, virulence and stress response. *International journal of biotechnology for wellness industries*, 2(3):101.

[181] T. L. Raivio. Everything old is new again: an update on current research on the cpx envelope stress response. *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research*, 1843(8):1529–1541, 2014.

[182] Z. Li and B. Demple. Soxs, an activator of superoxide stress genes in escherichia coli. purification and interaction with dna. *Journal of Biological Chemistry*, 269(28):18371–18377, 1994.

[183] K. Trchounian and A. Trchounian. Escherichia coli multiple [ni–fe]-hydrogenases are sensitive to osmotic stress during glycerol fermentation but at different phs. *FEBS letters*, 587(21):3562–3566, 2013.

[184] S. E. Hanlon, J. M. Rizzo, D. C. Tatomer, J. D. Lieb, and M. J. Buck. The stress response factors yap6, cin5, phd1, and skn7 direct targeting of the conserved co-repressor tup1-ssn6 in s. cerevisiae. *PloS one*, 6(4):e19060, 2011.

[185] A. G. Leiro, S. R. Lombardero, A. V. Vázquez, M I. G. Siso, and M. E. Cerdán. The yeast genes rox1, ixr1, sky1 and their effect upon enzymatic activities related to oxidative stress. In *Oxidative Stress-Molecular Mechanisms and Biological Effects*. InTech, 2012.

[186] C. Rodrigues-Pousada, T. Nevitt, and R. Menezes. The yeast stress response: Role of the yap family of b-zip transcription factors the pabmb lecture delivered on 30 june 2004 at the 29th febs congress in warsaw. *The FEBS journal*, 272(11):2639–2647, 2005.

[187] B. Mai and L. Breeden. Xbp1, a stress-induced transcriptional repressor of the saccharomyces cerevisiae swi4/mbp1 family. *Molecular and cellular biology*, 17(11):6491–6501, 1997.

[188] J. Lee, C. Godon, G. Lagniel, D. Spector, J. Garin, J. Labarre, and M. B. Toledano. Yap1 and skn7 control two specialized oxidative stress response regulons in yeast. *Journal of Biological Chemistry*, 274(23):16040–16046, 1999.

[189] C. Mascarenhas, L. C. Edwards-Ingram, L. Zeef, D. Shenton, M. P. Ashe, and C. M. Grant. Gcn4 is required for the response to peroxide stress in the yeast saccharomyces cerevisiae. *Molecular biology of the cell*, 19(7):2995–3007, 2008.

[190] N. Masuda and N. Konno. Vip-club phenomenon: Emergence of elites and masterminds in social networks. *Social Networks*, 28(4):297–309, 2006.

[191] S. Zhou and R. J. Mondragón. Accurately modeling the internet topology. *Physical Review E*, 70(6):066108, 2004.

[192] S. Zhou and R. J. Mondragón. The rich-club phenomenon in the internet topology. *IEEE Communications Letters*, 8(3):180–182, 2004.

[193] P. Csermely, A. London, L. Wu, and B. Uzzi. Structure and dynamics of core/periphery networks. *Journal of Complex Networks*, 1(2):93–123, 2013.

[194] V. Colizza, A. Flammini, M A. Serrano, and A. Vespignani. Detecting rich-club ordering in complex networks. *Nature physics*, 2(2):110, 2006.

[195] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

VITA

Satyaki Roy was born in Serampore, West Bengal, India. He graduated with a Bachelor of Science (B. Sc.) in Computer Science in 2012 from St. Xavier's College, Kolkata, India with an academic grade *A*. Subsequently, he pursued a Masters (M.Sc.) in Computer Science from the same institution, and finished third in the order of merit in 2014. He joined Missouri University of Science and Technology, Rolla, USA as Ph.D. scholar in Computer Science in Fall 2014 under Dr. Sajal K. Das. During this time, he served as graduate teaching assistant in three graduate and undergraduate courses. In December 2019, he received his Ph.D. in Computer Science from Missouri University of Science and Technology.