Mobile Netw Appl (2016) 21:191–205
DOI 10.1007/s11036-016-0708-6

The Structural Role of Feed-Forward Loop Motif in Transcriptional Regulatory Networks

Bhanu K. Kamapantula[1] · Michael L. Mayo[2] · Edward J. Perkins[2] · Preetam Ghosh[1]

Published online: 28 March 2016

Abstract We present multiple approaches to identify the significance of topological metrics that contribute to biological network robustness. We examine and compare the communication efficiency of transcriptional networks extracted from the bacterium *Escherichia coli* and the baker's yeast *Saccharomyces cerevisiae* using discrete event simulation based *in silico* experiments. The packet receipt rate is used as a dynamical metric to understand information flow, while unsupervised machine learning techniques are used to examine underlying relationships inherent to the network topology. To this effect, we defined sixteen features based on structural/topological significance, such as transcriptional motifs, and other traditional metrics, such as network density and average shortest path, among others. Support vector classification is used with these features after parameters were identified using a cross-validation grid-search method. Feature ranking is performed using analysis of variance F-value metric. We found that feed-forward loop (FFL) based features consistently show up as significant in both the bacterial and yeast networks, even at different noise levels. We then use a supervised machine learning technique (random forests) to investigate the structural prominence of the FFL motif in information transmission using sub-networks (larger sample size compared to the unsupervised approach) extracted from *Escherichia coli* transcriptional regulatory network. Further, we study the role of FFLs in signal transduction within the complete *Escherichia coli* regulatory network. Although our work reveals a minimal role of FFLs in signal transduction, it highlights the structural role of FFLs in information transmission captured by random forest regression. This work paves the way to design specialized engineered systems, such as wireless sensor networks, that exploit topological properties of natural networks to attain maximum efficiency.

Keywords Biological robustness · Transcriptional network · Feed-forward loop · Signal transduction

1 Introduction

Many functional aspects of transcriptional networks appear to be preserved despite the presence of noise or other disruptions. For example, some bacteria have been shown to survive despite extensive 'rewiring' of their transcriptional network topologies [9]. In some cases, such robustness to function can be attributed to the network structure alone, owing to its power-law degree distribution [1]. In other cases, the abundance of highly repetitive subnetworks, termed network motifs [22], have been correlated with an ability of the system to persist in a dynamically stable state [20]. One interesting example of such a motif is the feed-forward loop (FFL)–a three-node subnetwork wherein the top-level protein regulates the expression of a gene via two paths, which appears to be more abundant in some

✉ Preetam Ghosh
pghosh@vcu.edu

Bhanu K. Kamapantula
kamapantulbk@vcu.edu

Michael L. Mayo
Michael.L.Mayo@usace.army.mil

Edward J. Perkins
Edward.J.Perkins@erdc.dren.mil

[1] Virginia Commonwealth University, Richmond VA, USA

[2] US Army Engineer Research and Development Center, Vicksburg MS, USA

transcriptional networks than found in randomized versions [22]. Indeed, FFLs have received much attention, due in part to their information-processing ability. For example, they have been reported to speed-up or slow-down response times without any feedback loop [16].

This ability to function despite experiencing significant disruptions to communication seems to be a generic property of biology [14], and finding general properties or 'laws' that can be used to engineer this feature into man-made systems remains an open challenge [6, 15]. We make headway toward this goal by using machine learning techniques to interrogate the relationship between topological and dynamical properties of transcriptional networks, but viewed from the angle of the application area; in this case, a wireless sensor network. Here, nodes with communication capacity may continually experience channel noise, which has parallels in molecular biology: proteins and other signaling biomolecules are continually made and destroyed, leading to uncertainty in the channel capacity of a signaling pathway. Our approach to this problem is to combine discrete event simulation and support vector machine learning techniques to identify important system features that contribute to the information flow across such networks. Discrete event simulation can capture dynamic behavior of the system by modeling information transmission as a set of independent events under custom perturbations using channel noise and congestion-based information loss; while machine learning techniques can be used to identify underlying patterns in the data.

The NS-2 framework simulates information flow across wireless networks in terms of packet transport; we employ it here to quantify dynamical network robustness by measuring the packet receipt rates at various destination nodes in the model networks. Packet receipt rate is the ratio of number of packets successfully received at sink/destination nodes to the number of packets sent by the source node(s). While biological systems do not strictly communicate using information packets, they do employ signal transduction pathways that can be thought of as a series of activation steps, depending on concentration thresholds. This analogy can be taken further, given that biology is often redundant, in the sense that many pathways may be activated to achieve a single goal, reminiscent of flooding. We have described such similarities in detail before [5, 10, 11]. This paper builds upon our previous work and explores properties crucial for robustness in transcriptional networks to design specialized wireless sensor network topologies.

The procedure followed here is described in Fig. 1. Section 2.1 describes the network extraction–as shown in Fig. 1 (Step 1)–from *E. coli* and Yeast model networks using the GeneNetWeaver software [21]. Section 2.2 details the simulation setup and the determination of robustness (Fig. 1 (Step 2)) using NS-2 software. Section 3 describes the support vector machine technique used to identify data patterns followed by the determination of labels using k-means clustering algorithm, and feature definition.

2 Methods

2.1 Model transcriptional networks

The GeneNetWeaver software [21] is used here to extract subnetworks from transcriptional network datasets for the bacterium *Escherichia coli* and the common baker's yeast *Saccharomyces cerevisiae*. One hundred networks each of five different network sizes $n = 100$, 200, 300, 400, and 500, where n is the number of nodes, were considered. For simplicity, we will refer to networks derived from *S. cerevisiae* as 'Yeast' networks, whereas the bacterial networks will be referred to as *E. coli* networks. For our purposes, we map the transcription factors as transmitting/forwarding nodes, the genes as sinks while the edges represent interactions between participating nodes; thus, we ignored the regulatory type of each link.

2.2 Simulation setup

Network simulator (NS-2) [18] is used here to simulate packet transmissions in the mapped network. Nodes corresponding to genes that code for transcription factors are taken as the source nodes, whereas nodes corresponding to non-regulating genes are considered to be the sink nodes. While source nodes can send and forward packets, sink nodes may only receive packets without forwarding them onto others. We adopt a flooding type protocol, wherein each non-sink node may forward the received packets to its outgoing edges.

To account for noise, three different loss scenarios are considered, in which up to 20 %, 35 % and 50 % of packets can be *lost* in transit. This affects the packet receipt rate, which is determined to be the ratio of number of packets received at all sinks to the number of packets transmitted by source nodes, which we represent as a percentage of the total sent packets. This dynamical system is perturbed by fluctuating the loss level. Since the simulation setup considers channel fluctuation and congestion-based perturbations, we consider a network more *robust* when it exhibits a higher level of packet receipts.

2.3 Motif structural redundancy and packet receipt

What is the impact of structural redundancy, contributed by transcriptional motifs like FFLs (e.g. Fig. 2 (b)(1)), on the information flow (packet transmission) through a complex network? To examine this, we first tracked and identified all

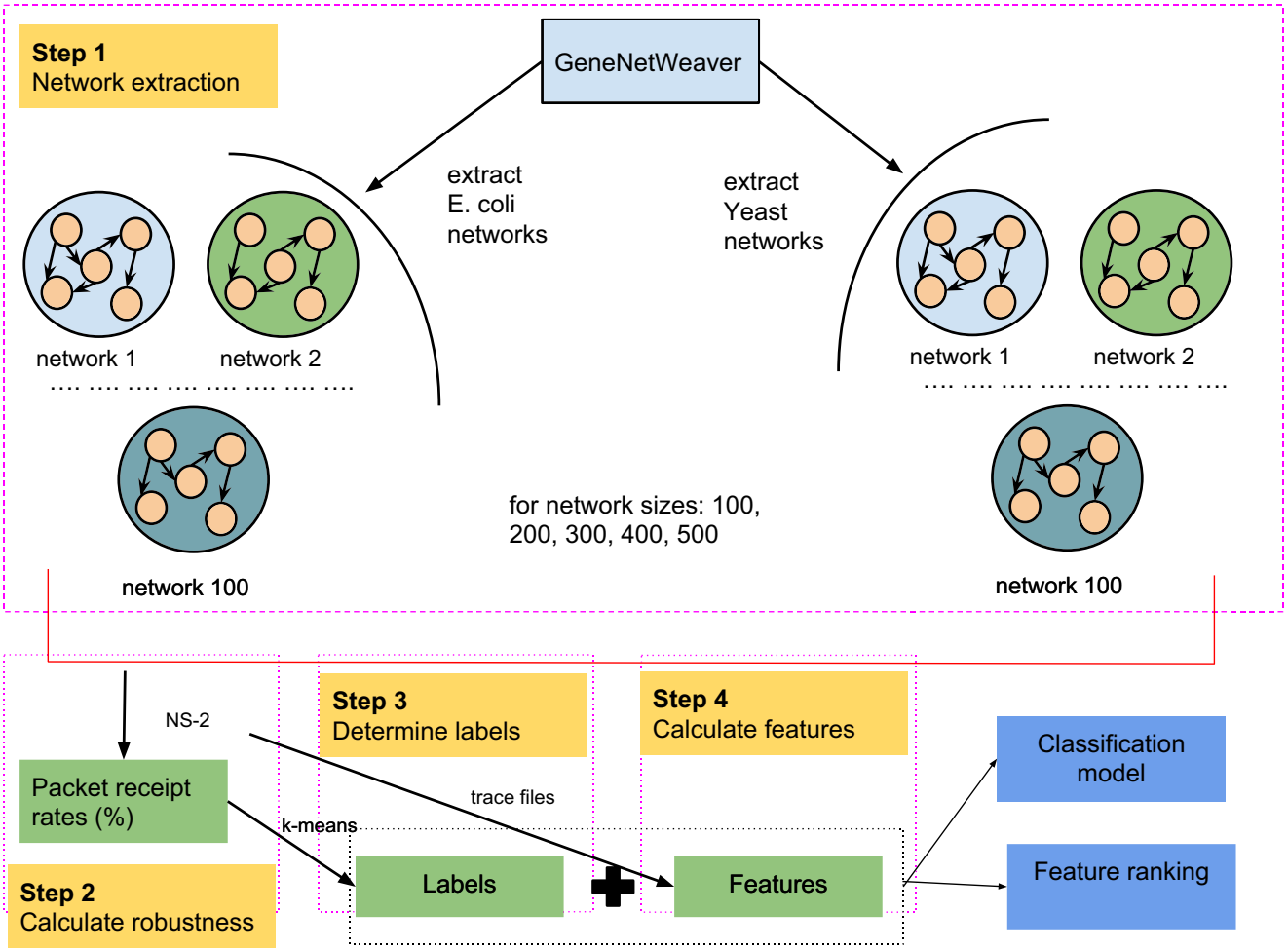


Fig. 1 Schematic of the procedure followed in this work

paths (node-hops) traveled by successfully received packets. We then used this history to identify all FFLs that possess a nonempty intersection with these successful paths.

3 Support vector machine modeling

Among several others, support vector machine (SVM) is a supervised machine learning (ML) technique used for classification of data [7]. Our goal here is to use SVMs to first identify, and then to determine, which topological features of transcriptional networks best capture the robustness of a network. An SVM model identifies a *classifier* (boundary that separates data) which best classifies the given data. While linear classifier suits well in some instances, others may require non-linear separation boundaries. The implementation of such linear or non-linear boundaries in an SVM model is achieved using kernel functions; possible kernel functions include: linear, polynomial, radial basis function (RBF) and sigmoid. An SVM model predicts the target value of the test data given the features.

A schematic of the SVM dataset is shown in Fig. 2a. It contains a set of instances, that are combinations of labels and features. The term *label* is attributed to an output which describes a feature, which is a property of the dataset. In addition, each feature is assigned a unique ID. For example, we employed ten datasets, which constitute five sampled subnetworks each from the datasets for E. coli and yeast. Each of these five datasets corresponds to a particular network size, as measured by the number of nodes, i.e. $n = 100, 200, 300, 400,$ or 500. One hundred networks were sampled from the source datasets for each size, and each such sampled subnetwork is an example of an *instance*.

We used Python and *scikit-learn* package [19] to identify features and build SVM classification models. *scikit-learn* utilizes the popular ML libraries *libsvm* and *liblinear*. We follow the data preprocessing and model selection steps from [8]. We perform data scaling after feature determination (Section 3.5) and a grid search (Section 3.4) to identify best parameters to classify data. Our goal is two-fold: a) to build a classification model and

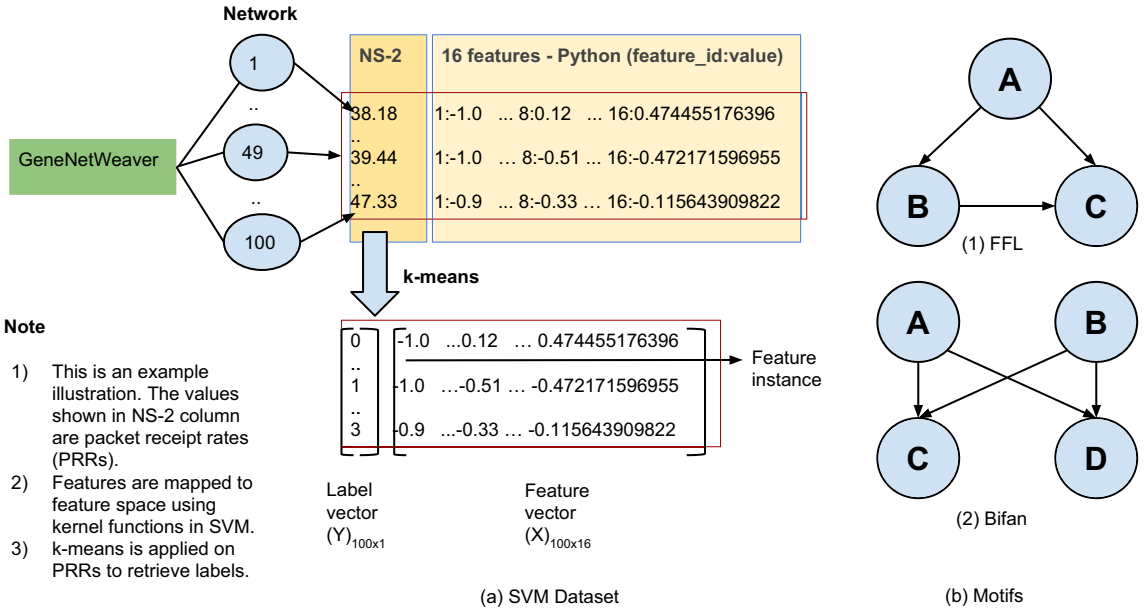


Fig. 2 (**a**) SVM Dataset for each network size, at specific perturbation level (**b**)(1) FFL, (**b**)(2) bifan motifs

b) rank features. Feature ranking is performed using analysis of variance F-test which does not use the model created by SVM.

3.1 Assigning labels for SVM

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

3.2 Data pruning

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

3.3 Training and testing

The pruned data is used as training and testing sets for the SVM models. Each dataset is split into 75 % training and 25 % testing sets. In order to avoid overfitting the data, 5-fold cross validation is used to randomize the 75/25 split into training/testing datasets. In a 5-fold cross validation test, the split is performed five different times; labels are stored in a vector, and corresponding feature instances are stored in another, different vector. Continuing the example stated in the Section 3.2, now the training set contains $\{1 : 27, 0 : 26, 3 : 21\}$ and the testing set contains $\{1 : 10, 0 : 8, 3 : 7\}$.

3.4 Parameter selection

A grid search is performed to identify the 'best' parameter set in which to build an SVM model. Grid search uses *k*-fold cross validation and builds a classifier for each set of parameters. Each classifier is then tested using the $F1$ *score*, which is a weighted average of precision and recall [19]. The set of parameters used are shown in Table 1. C is the

Table 1 Grid search parameters identified using the cross validation method described (20 % perturbation)

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

regularization constant and γ is a kernel hyper-parameter. 1, 10, 100, 1000 are used as C values for Linear, RBF, Polynomial kernels. The set of values 0.0001, 0.001, 0.01, 0.1, 1 and 2 are used as γ for RBF kernel. A γ value of 1 is used for polynomial kernel. 1, 2, 3, 4, 5 are used as $degree$ values (applicable only to Polynomial kernel). Large C overfits the data (high cost for misclassification) while large γ in polynomial kernel ensures a smoother decision boundary.

3.5 Features

ML techniques use underlying properties, referred to as features, of the data to describe relationships. For each data instance, features are mapped to corresponding labels, as described below. Given a network graph, $G(V, E)$, wherein V is the set of supporting vertices, and E is the set of edges linking those vertices, we define the following SVM features: features defined based on the network topology are given in Sections 3.5.1 to 3.5.11, whereas features defined in terms of NS-2 simulation traces are given by Sections 3.5.12 to 3.5.13. These latter features are hereafter referred to as 'path-based features'. In total, sixteen features are studied; all features/metrics are normalized to the interval $[-1, 1]$ to remove any artificial bias. The normalization was done as follows:

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

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

3.5.1 Network density

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

$$ND = \frac{|E|}{|V|\,(|V|-1)}. \tag{2}$$

3.5.2 Average shortest path

The average shortest path (ASP) of a network is the shortest of all path-lengths, $\min\{d\,(V_1, V_2)\}$, measured between any two network nodes V_1 and V_2. This metric captures the ability of two nodes to communicate and is given by:

$$ASP = \sum_{V_1, V_2 \in V} \frac{\min\{d\,(V_1, V_2)\}}{|V|\,(|V|-1)}. \tag{3}$$

3.5.3 Degree centrality

Degree centrality of a node (n_{dc}) is defined as the number of edges incident on the node. Thus, it provides a measure of reception to others within a network and is given by:

$$n_{dc} = \frac{deg(n)}{|V|-1} \tag{4}$$

wherein n denotes the node and $deg(n)$ is its degree. In order to identify the impact of genes, which are regulated by transcription factors, the collective average degree centrality of genes (ADCG) is considered as a feature, along with average degree centrality of the network (ADC).

3.5.4 Transcription factor percentage

Transcription factor percentage (TFP) is a measure of the fraction of networked nodes that serve as transcription factors which regulate genes and is given by:

$$TFP = \frac{|V_{TF}|}{|V|}, \tag{5}$$

wherein $|V_{TF}|$ is the total number of transcription factors in the network.

3.5.5 Genes percentage

In complement to TFP metric, Eq. 5, we define the genes percentage (GP) as the fraction of networked that can be identified as genes. This quantity can be calculated with the equation:

$$GP = \frac{|V_G|}{|V|}, \tag{6}$$

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

3.5.6 Source to sink edge percentage

Larger networks are more likely to support links that directly connect source to sinks within the network, facil-

itating information flow. Thus, we propose a metric that quantifies this property: the source to sink edge percentage (SSEP), which we define as the fraction of direct edges, $|E_{SS}|$, from source nodes to sink nodes compared to the total number of edges in the network:

$$SSEP = \frac{|E_{SS}|}{|E|}. \tag{7}$$

3.5.7 FFL abundance

FFL abundance (FFLD) is the ratio of total edges in the network that intersect with edges from at least one FFL to the total edges in the network, and is given by:

$$FFLD = \frac{|E_{FFL}|}{|E|}, \tag{8}$$

where E_{FFL} : number of edges participating in FFLs.

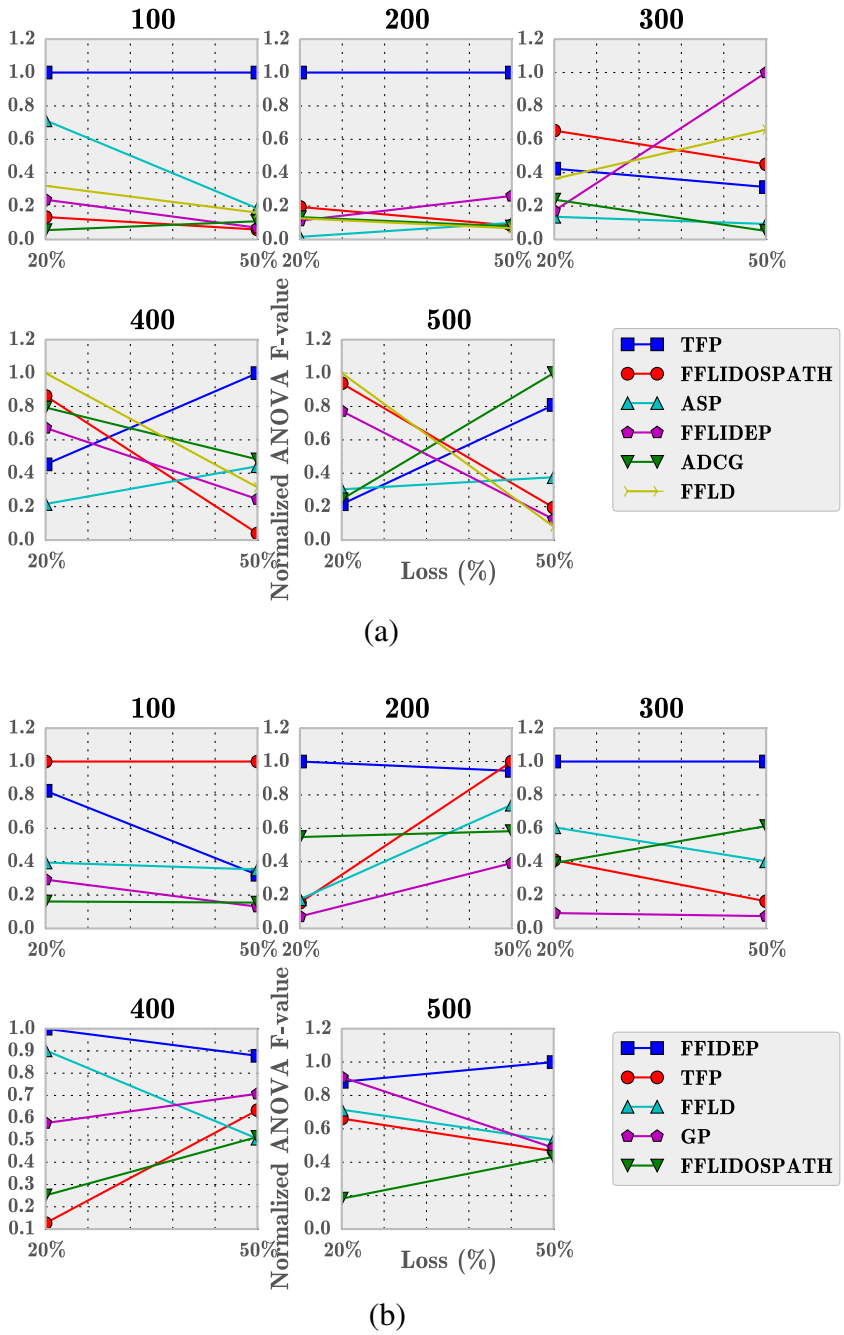


Fig. 3 Variation of top 5 features in each E. coli (panel (a)) and yeast (panel (b)) networks, at losses 20 % and 50 % (Sizes = 100, 200, 300, 400, 500)

3.5.8 FFLDED

Figure 2(b)(1) illustrates an FFL, which is hierarchical and composed of two regulatory paths. The first is a 'direct' linkage from nodes A to C, whereas an 'indirect' path accounts for regulation of node C through node B. Here, the FFL direct-edge density (FFLDED) is the ratio of FFL direct edges, $|E_{FFLDE}|$, to the total edges in the network, and is given by:

$$FFLDED = \frac{|E_{FFLDE}|}{|E|}. \quad (9)$$

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

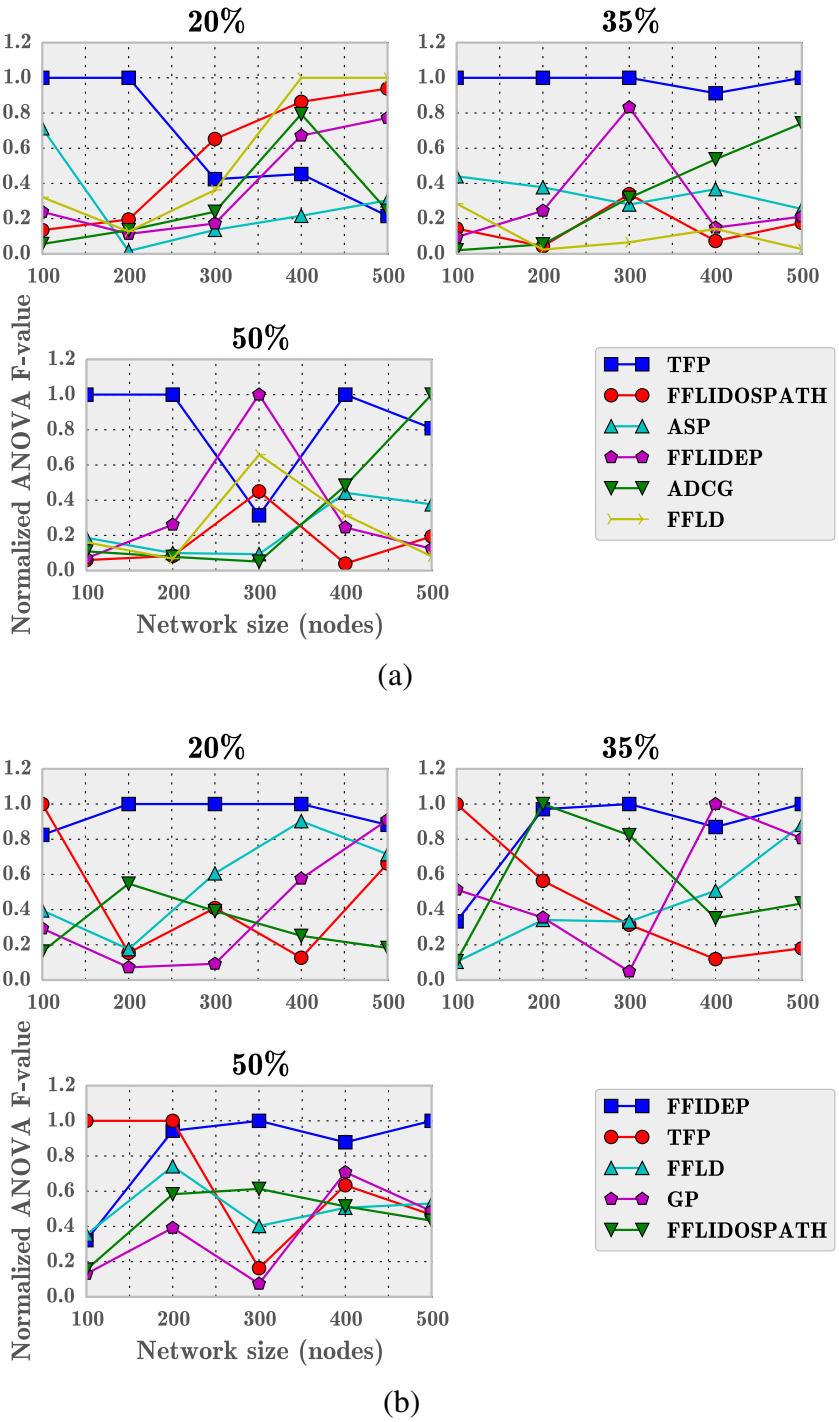


Fig. 4 Variation in normalized ANOVA F-values for the top 5 features in each *E. coli* (panel (a)) and *yeast* (panel (b)) networks, at losses 20 % and 50 % (Sizes = 100, 200, 300, 400, 500)

3.5.9 FFLSSPD

The FFL source to sink edge density (FFLSSPD), is the fraction of direct source-sink edges that are also part of an FFL to the total number of source-to-sink edges in the network. This metric decouples the influence of FFLs from all other source-to-sink edges in the network.

3.5.10 FFLDEP

The FFLDED metric above accounts for the fraction of direct-edge FFL links present in the network. However, a single link may potentially appear more than once if it is 'shared' among two or more FFLs. We define a separate measure that ignores multiple copies of any single link, which is given by:

$$FFLDEP = \frac{|E_{FFLDE}|}{|E|}, \tag{10}$$

wherein $|E_{FFLDE}|$ is the number of unique direct-edges in FFLs embedded in the network.

3.5.11 FFLIDEP

Indirect FFL edge percentage (FFLIDEP) is the ratio of the number of unique FFL indirect edges to the total number of sequential, two-step paths in the network. Thus, it is similar to FFLDED, but measured against the indirect edge of the FFL as follows:

$$FFLIDEP = \frac{|E_{FFLIDE}|}{|E_{TEP}|}, \tag{11}$$

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

3.5.12 Direct-edge trace participation

Each NS-2 simulation results in a set of 'traces' that map packet-transport histories for packets sent and received successfully from source to sinks. Similar to Eq. 9, but considering packet trace history, we measure the ratio of the number of unique FFL direct edges that participate in successful packet paths to the total number of unique FFL direct edges, termed FFLDSPATH. Another feature FFLDOSPATH can be defined similar to FFLDSPATH if we allow for duplication of FFL direct-edges; this metric allows for FFL direct edges to participate multiple times in successful packet delivery.

3.5.13 Indirect-edge trace participation

Finally, we measure the ratio of the number of unique active FFL indirect edges that participate in successful packet trace histories to the number of unique FFL indirect edges. This

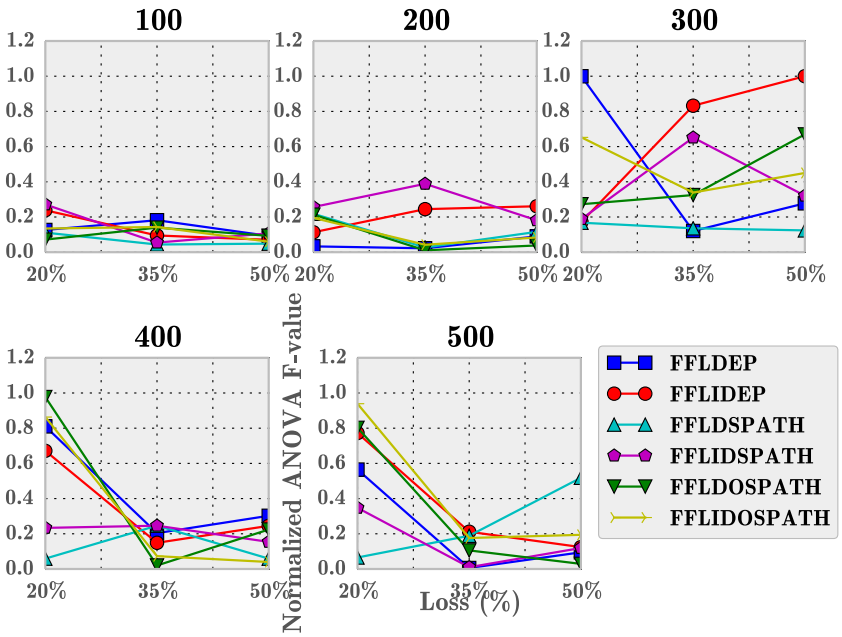


Fig. 5 Variation of FFL participating direct and indirect edge-based features at 20 %, 35 % and 50 % loss for E. coli networks (Sizes = 100, 200, 300, 400, 500)

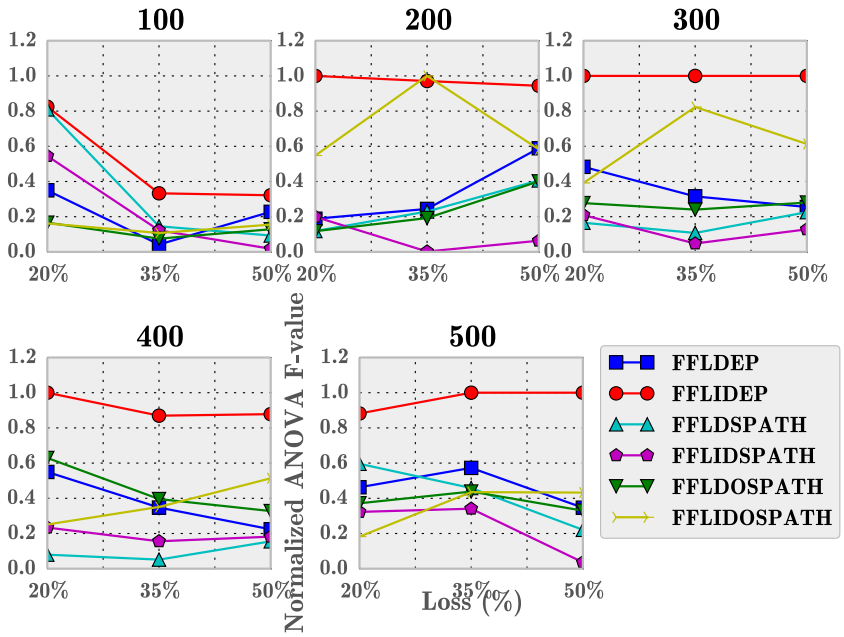


Fig. 6 Variation of FFL participating direct and indirect edge-based features at 20 %, 35 % and 50 % loss for Yeast networks (Sizes = 100, 200, 300, 400, 500)

metric is termed FFLIDSPATH. Similarly, we allow for the multiple counting of a single FFL indirect path in the contribution to successful packet trace history. This metric is termed FFLIDOSPATH; i.e., FFL indirect edges can be leveraged more than once to successfully deliver a packet.

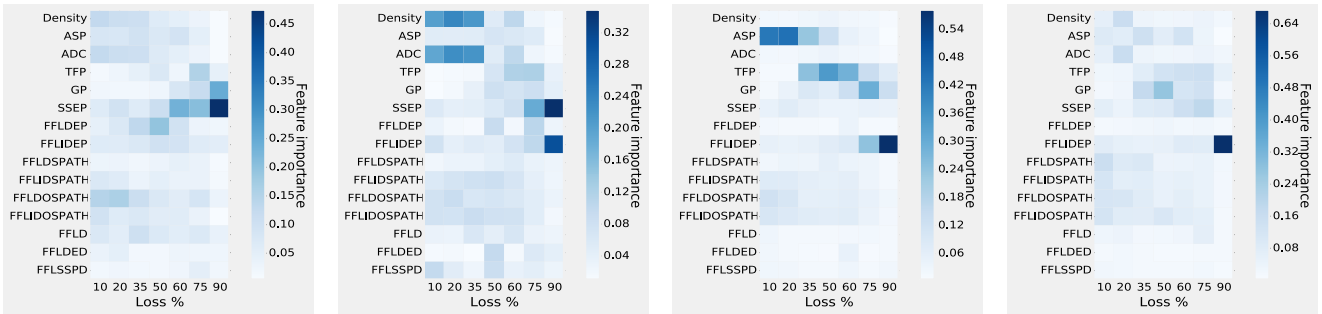
3.6 Feature ranking

The identified features are ranked using the analysis of variance (ANOVA) F-value metric. This metric compares the inter-class variance to intra-class variance [19]. A higher F-value denotes higher significance of a feature. F-value captures feature significance individually but not the mutual feature dependence.

4 Important features

4.1 Packet receipt rates in transcriptional networks

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



(a) 100 nodes (b) 200 nodes (c) 300 nodes (d) 400 nodes

Fig. 7 Feature importance for different network sizes and perturbation levels (loss %); each cell is an average of 100 runs. Feature importance ranges from 0 to 1. Higher feature importance value signify higher importance

sequent decrease in received packets, independent of the global network topology.

4.2 Feature ranking in transcriptional networks

In the NS-2 simulations, channel noise and congestion based packet drops account for internal perturbations. As mentioned above, fluctuation in packet loss (%) is considered as a perturbation/stressor to the information flow. This channel loss stressor is used in the SVM models to explore the significance of motifs on structural redundancy and packet receipt rates.

4.2.1 Top-ranking features

Fifteen different SVM models, one for each pair of network size and perturbation level, are used to select features/metrics for one specific type of transcriptional network. Let us examine the feature selection in *E. coli* networks for one of the fifteen SVM model instances. For each network size, the top five features are selected, according to the criterion that each the most 'influential' features should occur at least three times in the top five features as scored across different network sizes. For *E. coli* networks, this top-ranked set is given by the features: TFP, FFLIDOSPATH, ASP, FFLIDEP, ADCG, FFLD (Fig. 3a). Similarly, features so identified from yeast networks are: FFLIDEP, TFP, FFLD, GP, FFLIDOSPATH (Fig. 3b). All influential features identified from the SVM models in terms of packet receipt rates relate to the FFL motifs.

4.2.2 Feature stability at different perturbation levels

As a preliminary experiment, we tested the prevalence of transcriptional network features at different noise perturbation levels. Here, our intention is to assess if structural or dynamic features prevail in feature significance. The result of this on *E. coli* networks is shown in Fig. 3a[1] and on Yeast networks in Fig. 3b. FFLIDEP ranks consistently higher in most cases (except at network size 100) than other features. Similarly, FFLD and GP rank in the top two or three at different network sizes. An interesting observation is that three (FFLIDEP, FFLD, FFLIDOSPATH) out of five top ranked features are related to FFLs.

4.2.3 Feature ranking across different network sizes

We next investigate if the relative importance of features vary across different network sizes. Figure 4a shows that in *E. coli*, TFP ranks consistently stable in most cases at 35 % and 50 % perturbation levels (except at network size 300). FFLIDOSPATH, FFLD and FFLIDEP rank higher in some instances. Figure 4b shows the relative importance of features in Yeast. Here, FFLIDEP is relatively stable across different network sizes while FFLD and GP are stable at some cases but not conclusively overall. Hence, a combination of conventional metrics (GP and FFL-derived features) can be used to engineer networks that are robust to perturbation.

4.2.4 Comparison of FFL based features

The results from the above two studies motivate us to investigate the variation of FFL based features only instead of the top five identified features. A general trend can be observed from Fig. 5 that FFL-based features have higher significance (based on normalized ANOVA F-value) from network sizes 300 and above. Also Fig. 5 shows that FFLIDEP is ranked first among the six FFL based features in certain instances (100, 200, 300 and one instance in 400, 500 network sizes). Figure 6 shows the ranking for Yeast networks. FFLIDEP ranks the highest for all network sizes and perturbation levels. Correlation between FFLDSPATH and FFLDOSPATH (derived from FFLDSPATH) is not always proportional suggesting that there is more to FFL participation than the number of successful FFL direct path contribution; the position of FFLs in the network might also play a critical role. FFLDEP, FFLIDEP and FFLIDOSPATH consistently rank as the top three features at different perturbation levels. This directly reveals the importance of the percentage of FFL direct edges present in the network and the number of times those edges were used in successful packet transmissions.

5 Role of FFLs in information transmission

Our goal here is to study the structural role of FFLs in information transmission. Specifically, does the indirect path ($A \rightarrow B \rightarrow C$ in Fig. 2b; FFLIDEP is determined based on such paths) in FFLs stand out when the network is perturbed? Our assumption being is that when a network is perturbed, direct FFL edges ($A \rightarrow C$ in Fig. 2b: FFLDEP is determined based on such edges) are destroyed and information transmission occurs via the indirect path. In order to study this feature importance, we use random forest regression.

In Section 3.1 (followed in [12] and [13]), we used k-means to cluster packet receipt rates and identified important features using F-ANOVA metric after performing SVM classification. In a significant improvement in methodology, we modify the problem from an unsupervised technique (to cluster output labels) to supervised regression problem as it is more suited to the context of the problem we are

[1] For the figure to be legible, X and Y labels are displayed only once. This is done for Figs. 3a – 6.

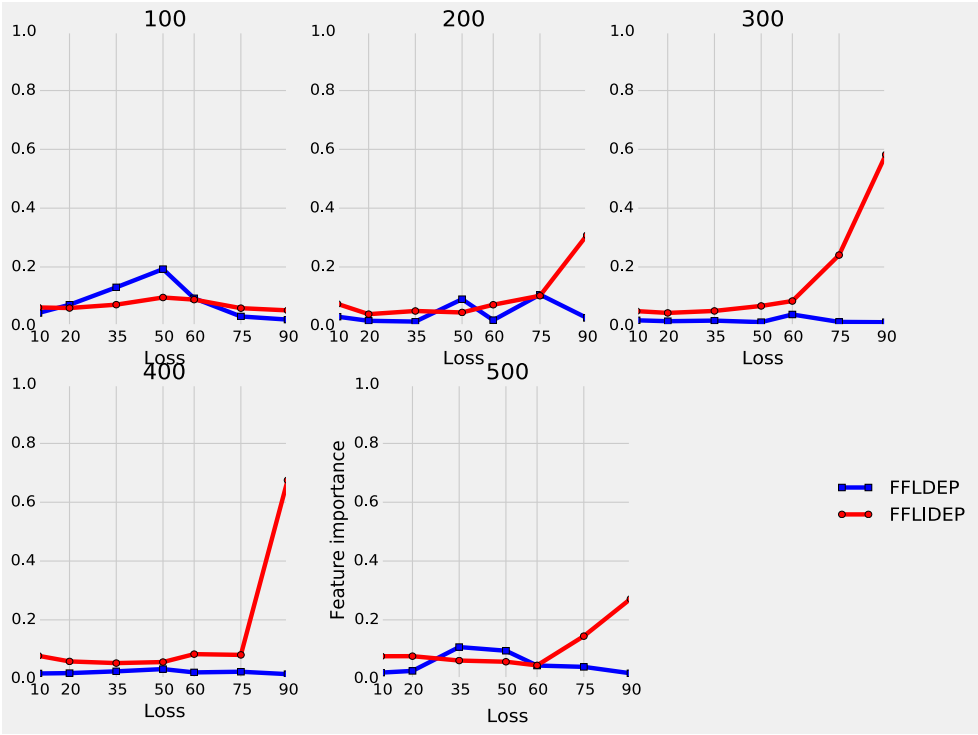


Fig. 8 Relative importance of FFLDEP vs FFLIDEP at different perturbation levels (loss %) and network sizes. Each data point is an average of 100 runs

trying to address. Specifically: 1) given a sample set, can we effectively determine feature (input for any network) importance that impacts biological network robustness (network output)? 2) training a model on a given a sample set, how effectively can we predict robustness values for a new network? We explore these questions next.

5.1 Data

In order to comprehensively analyze E. coli subnetworks, we increased the sample size from 100 to 1000 for each network size (100, 200, 300, 400 and 500). After pruning out the networks that are disconnected, we ended up with 974, 943, 957, 932 and 941 samples for network sizes from 100 to 500 respectively. We also increased the number of perturbation levels from 3 (20 %, 35 % and 50 %) to 7 (10 %, 20 %, 35 %, 50 %, 75 % and 90 %). This helps to better understand which features are prominent at varying network perturbation levels. Feature correlation was performed to prune the feature set from 16 (from Section 3.5 and [12, 13]) to 15 in this experiment. We also changed the feature scaling from [-1, 1] (in Section 3.5) to [0, 1] for this study.

5.2 Random forests

We experimented with various machine learning strategies including randomized PCA, LDA, linear regression and recursive feature elimination. Randomized PCA could not exploit the output label (network packet receipt rate in our case) data to minimize feature space. LDA performed poorly and linear regression models such as LASSO and ElasticNet yielded poor coefficient of determination values. For feature ranking, recursive feature elimination techniques were tried; here, each feature is removed and model performance is estimated. The model most impacted (negatively) reveals the most important feature. These techniques did not yield good coefficient of determination values either which prompted us to explore random forests (RF) [3] to identify best network characteristics.

RF is an ensemble approach to solve classification and regression tasks and uses several trees (estimators) to predict the outcome of test data. A tree is constructed from sample data filtered from the training dataset. At each terminal node of the tree, m features are selected out of the feature set and a best feature is identified where the tree is split into child nodes. This is repeated until the selected sample size from the training data is the least. By using several trees and averaging the predictions, the variance across the trees is reduced. Then, mean squared error (MSE) is used to determine the best of estimators to build the RF prediction model. We tested 19 different estimators (10 to 100 in steps of 5) to build the RF models; the one with least MSE is selected to calculate feature importance. The entire RF algorithm is executed for 100 runs to negate variations in feature importance calculation due to randomization. Hence, at each run, the following are calculated: MSE (best out of 19 runs), feature importance and coefficient of determination (using the model with least MSE out of 19 estimators). The importance

of each feature is calculated by averaging the total reduction in node impurity (as in scikit toolkit [19]) across all estimators. The feature importance reported in Fig. 7a, b, c and d (for different network sizes) are an average of all the 100 runs. Seven different RF learning models are implemented in each of these Figures. A total of thirty five different models (seven RF models for each network size at five network sizes) are used to derive the conclusions. Figure 7 shows that the features which stand out vary from one network size to the other and one perturbation level to another.

5.3 Feature importance

It can be observed that FFLIDEP emerges as a strong feature at higher loss. Features like Density, TFP and GP show up to be more important than others in multiple instances. We can recollect that FFLIDEP is the percentage of indirect FFL edges that are present in the network compared to the total edges. Our primary hypothesis is to test if the reduction in feature importance of FFLDEP at higher loss leads to higher importance of FFLIDEP. We explicitly observe this in Fig. 8; this hypothesis is actually true for all network sizes except 100. Hence, we intended to scan the entire E. coli regulatory network for these patterns and introduce a way to understand FFL distribution as described below.

5.4 Model prediction

In order to measure the performance of the RF model, we use a standard metric—coefficient of determination (COD). COD measures the predicted value performance against real value (packet receipt rate in our case). Good regression models will have a COD value closer to 1 and poor models will have a value closer to 0. Figure 9 present COD before (15 features described earlier) and after (number of features varies according to the model) feature reduction. For feature reduction, we first deduce feature importance and retain all features that have higher than average feature importance in a particular model. Figure 9 shows that feature reduction does not impact COD in majority of cases; hence, we retain models developed without feature reduction.

5.5 Signal transduction

Figure 10 is used as a reference to explain the following concepts. We consider FFLs that follow, what we term, a shortest path switch. Shortest path switch in FFLs is defined as follows: shortest path from node A to node C is always via the direct edge. However, under high noise the FFL direct edge may potentially be destroyed making the information flow from node A to node C occur via the indirect path (via node B). Note that this path switch only happens because there were no other shortest paths from node A to the sink other than the indirect $A \rightarrow B \rightarrow C$ path, which in turn implies, that the shortest path from A to the sink has

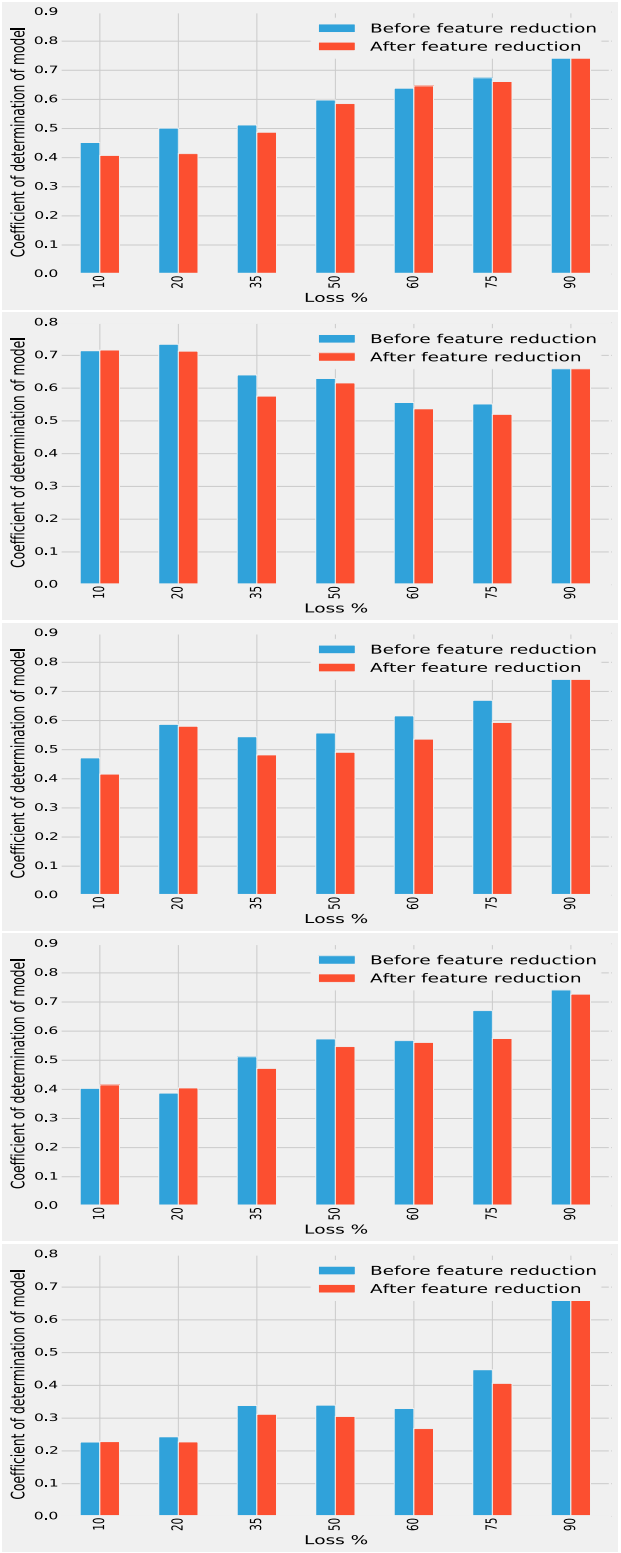


Fig. 9 COD before and after feature reduction using RF model for $n = 100$, 200, 300, 400 and 500 respectively

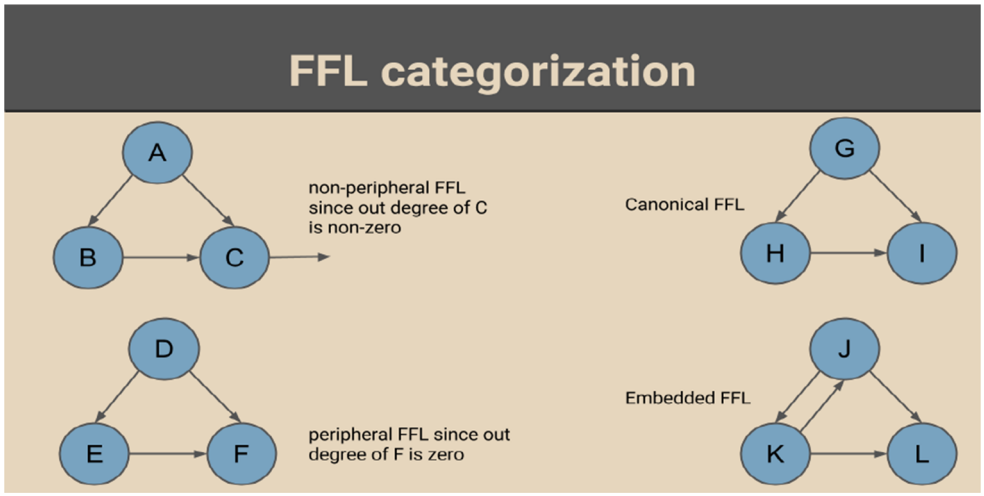


Fig. 10 Categorization of FFLs into peripheral/non-peripheral and embedded/canonical types

increased by one-hop. Here we identify all FFLs that switch the shortest path from the direct to the indirect FFL path.

We first group FFLs into two categories: canonical and embedded. FFLs with no additional edges among the nodes are considered to be canonical. FFLs with additional edges among the nodes are considered to be embedded. This is illustrated in Fig. 10. Further, we group each of these FFL categories into peripheral and non-peripheral FFLs. Peripheral FFLs are the ones in which the node being transcribed has no out degree, while in non-peripheral FFLs the nodes being transcribed have non-zero out degree.

Our interest to study FFLDEP and FFLIDEP in particular is due to the fact that these two FFL-derived features effectively capture the FFL path switch from direct to indirect for information transmission. Our idea to identify FFLs that are central to the E. coli network led to the study of the distribution of peripheral and non-peripheral categories of canonical and embedded FFLs. First, we identified all canonical and embedded FFLs. A majority of FFLs (64.5 % and 80.5 % respectively) switched paths due to the direct edge deletion in canonical and embedded FFLs. Figure 11 presents the detailed distribution of the physical location of FFLs within the E.coli network. This reveals that only a small number of FFLs (6 canonical non-peripheral and 26 embedded non-peripheral FFLs) participate in signal transduction within this network. Effectively, only the gene nodes in these filtered FFLs have outgoing edges enabling them to participate in signal transduction. Controlling the nodes in these filtered FFLs can establish critical patterns prevalent in the regulatory network.

Our next objective is to explore the significance of FFLs in signal transduction within the entire E. coli network. Signal transduction can be explained as follows: in a transcriptional network, the nodes (genes or transcription factors) that are regulated by transcription factor nodes could influence other nodes. In the context of information networks, this translates to the following: nodes that receive information can forward the data to other nodes. Following our FFL categorization, in order to understand the location of the FFLs responsible for signal transduction, we observe

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

Fig. 11 Distribution of categorized FFLs in E. coli. Column 2 presents the number (and the relative percentage) of respective FFL category. Column 3 presents the number of respective FFLs with path switches. In other words, we first count all the shortest paths from source to sinks that pass through the direct FFL edge ($A \rightarrow C$ in Fig. 2). Then, we knockout the direct edge and determine the number of shortest paths that involve the indirect FFL path ($A \rightarrow B \rightarrow C$ in Fig. 2). All such FFLs with indirect path switches are presented in Column 3

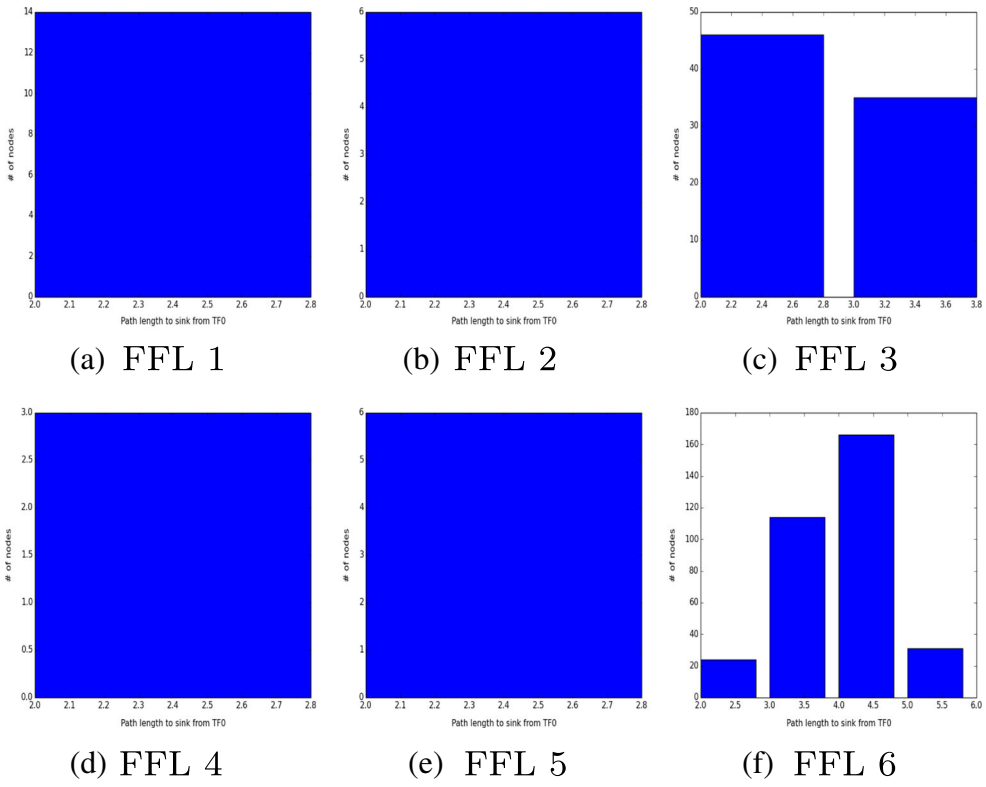


Fig. 12 The distribution of 6 canonical non-peripheral FFLs: number of hops to nearest sink nodes from topmost transcription factor (TF0, equivalent to *A* in Fig. 2b) in an FFL. X-axis represents the number of hops it takes to reach sink node. Y-axis represents the number of nodes that can reach the sink node. For example, in (f) there are 20 nodes that can reach their sink nodes within a distance of 2 hops

the distribution of six canonical non-peripheral FFLs (row 3, column 2 from Fig. 11) within the E. coli network. This is represented in Fig. 12 [2]. To this end, for each such FFL, we determine shortest paths from the topmost FFL node (equivalent to *A* in Fig. 2b, assumed to be a transcription factor) to the nearest sink node. We can notice in Fig. 12 that only two FFLs (c, f) make an impact in signal transduction.

6 Discussion & conclusion

We reported two machine learning approaches to understand the contributing factors to biological network robustness. Our experiments revealed the importance of modeling the problem of network robustness prediction as a supervised regression learning technique (random forests). This work improves the framework required to capture biological network robustness from the perspective of information transmission at different perturbation levels using which important network characteristics could be determined.

Building on this research, engineered networks can be created that are robust under lossy conditions using the features identified as important in this work. Researchers have proposed methods in the past to develop complex systems ensuring specific topological aspects-for instance, retaining overall degree distribution while growing networks in the Barabasi-Albert preferential attachment model [2]. Such models can be improved to generate complex networks by including the FFL derived features to ensure robust system behavior [17]. These features include average shortest path and the feed-forward loop motif. Each such structure contains two edge disjoint paths for a transcription factor node to regulate a gene node. This structure plays a prominent role in information forwarding at high perturbation levels. Most canonical FFLs exhibit only a single shortest path to their sink that passes through FFL direct edge; under noise when this direct edge becomes unavailable, information transport switches to the indirect FFL path which alternately suggest that the shortest path to sink for this FFL has increased by one-hop. While this work only explores the structural contribution of FFLs, other motifs (bifan, for instance) have also been shown to be promising in their contribution to biological network robustness. Recent research has also highlighted bow-tie motifs that play a role in biological signalling and information processing [4]. These motifs can introduce interesting dimensions to our work by adding new feature types to the network topologies.

Acknowledgments This work was partially funded by the US Army's Environmental Quality and Installations 6.1 basic research program. Opinions, interpretations, conclusions, and recommendations are those of the author(s) and are not necessarily endorsed by the U.S. Army. The work was also partially supported by the NSF.

[2] Similar figures for 26 embedded non-peripheral FFLs are not presented here

References

1. Albert R, Jeong H, Barabasi A (2000) Error and attack tolerance of complex networks. Nature 406:378
2. Barabási AL, Albert R (1999) Emergence of scaling in random networks. Science 286(5439):509–512
3. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
4. Friedlander T, Mayo AE, Tlusty T, Alon U (2015) Evolution of bow-tie architectures in biology. PLoS Comput Biol 11(3):e1004,055–e1004,055
5. Ghosh P, Mayo M, Chaitankar V, Habib T, Perkins E, Das SK (2011) Principles of genomic robustness inspire fault-tolerant wsn topologies: a network science based case study. In: 2011 IEEE international conference on Pervasive computing and communications workshops (PERCOM workshops), pp 160–165. IEEE
6. Ghosh S, Ghosh P, Basu K, Das SK (2005) Gama : an evolutionary algorithmic approach for the design of mesh-based radio access networks. In: 2005 IEEE international conference on Local computer networks, pp 374–381. IEEE
7. Hearst MA, Dumais S, Osman E, Platt J, Scholkopf B (1998) Support vector machines. IEEE Intelligent Systems and their Applications 13(4):18–28
8. Hsu CW, Chang CC, Lin CJ et al (2003) a practical guide to support vector classification
9. Isalan M, Lemerle C, Michalodimitrakis K, Horn C, Beltrao P, Raineri E, Garriga-Canut M, Serrano L (2008) Evolvability and hierarchy in rewired bacterial gene networks. Nature 452:840
10. Kamapantula BK, Abdelzaher A, Ghosh P, Mayo M, Perkins E, Das SK (2012) Performance of wireless sensor topologies inspired by e. coli genetic networks. In: 2012 IEEE international conference on Pervasive computing and communications workshops (PERCOM workshops), pp 302–307. IEEE
11. Kamapantula BK, Abdelzaher A, Ghosh P, Mayo M, Perkins EJ, Das SK (2014) Leveraging the robustness of genetic networks: a case study on bio-inspired wireless sensor network topologies. Journal of Ambient Intelligence and Humanized Computing:1–17
12. Kamapantula BK, Mayo M, Perkins E, Abdelzaher AF, Ghosh P (2014) Feature ranking in transcriptional networks: packet receipt as a dynamical metric. In: Proceedings of the 8th international conference on bioinspired information and communications technologies, pp 1–8. ICST
13. Kamapantula BK, Mayo M, Perkins E, Ghosh P (2014) Dynamical impacts from structural redundancy of transcriptional motifs in gene-regulatory networks. In: Proceedings of the 8th international conference on bioinspired information and communications technologies, pp 199–206. ICST
14. Kitano H (2007) Towards a theory of biological robustness, vol 3
15. Liu YY, Slotine JJ, Barabási AL (2011) Controllability of complex networks. Nature 473(7346):167–173
16. Mangan S, Alon U (2003) Structure and function of the feed-forward loop network motif. Proc Natl Acad Sci 100(21):11,980–11,985
17. Mayo M, Abdelzaher A, Perkins EJ, Ghosh P (2012) Motif participation by genes in e. coli transcriptional networks. Front Physiol 3(00357). doi:10.3389/fphys.2012.00357
18. McCanne S, Floyd S, Fall K, Varadhan K et al (1997) Network simulator ns-2. http://nsnam.sourceforge.net/wiki/index.php/Main_Page
19. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. J Mach Learn Res 12:2825–2830
20. Prill RJ, Iglesias PA, Levchenko A (2005) Dynamic properties of network motifs contribute to biological network organization. PLoS Biol 3(11):e343
21. Schaffter T, Marbach D, Floreano D (2011) Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. Bioinformatics 27(16):2263–2270
22. Shen-Orr S, Milo R, Mangan S, Alon U (2002) Network motifs in the transcriptional regulation network of escherichia coli. Nat Genet 31(1):64–68