Dynamical impacts from structural redundancy of transcriptional motifs in gene-regulatory networks

Bhanu K. Kamapantula [*]
kamapantulbk@vcu.edu

Michael Mayo [†]
Michael.L.Mayo@usace.army.mil

Edward Perkins [‡]
Edward.J.Perkins@usace.army.mil

Preetam Ghosh [§]
pghosh@vcu.edu

ABSTRACT

We examine and compare 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 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 carried out using 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 based features rank consistently high in both the bacterial and yeast networks, even at different perturbation levels. 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.

General Terms

Machine learning, feature ranking, complex networks, transcriptional networks

[*]Corresponding author - kamapantulbk@vcu.edu
Virginia Commonwealth University, 401 W Main St, Richmond, VA, USA

[†]Environmental Laboratory, US Army Engineer Research and Development Center, Vicksburg, MS 39180

[‡]Environmental Laboratory, US Army Engineer Research and Development Center, Vicksburg, MS 39180

[§]Virginia Commonwealth University, 401 W Main St, Richmond, VA, USA

.

Keywords

biological robustness, transcriptional network

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 [6]. In some cases, such a robustness to function can be attributed to the network structure alone, owing to its power-law degree distribution [1]. In other cases, the abundance of highly repetitive subnetworks, termed transcriptional motifs [17], have been correlated with an ability of the system to persist in a dynamically stable state [15]. One interesting example of a transcriptional motif is the feed-forward loop–a small, three-node subnetwork wherein the top-level protein regulates the expression of a gene via two paths, which appears to be more abundant in some transcriptional networks than found in randomized versions [17]. Indeed, feed-forward loops have received much attention, due in part to their information-processing ability. For example, they have been reported to speed-up or slow-down response times without any feedback loop [12].

This ability to remain useful despite experiencing significant disruptions to communication seems to be a generic property of biology [10], and finding general properties or 'laws' that can be used to engineer this feature into man-made systems remains a 'holy grail' of systems architecture and control theory [11]. We make headway toward this goal by using machine learning techniques to interrogate the relationship between topological and dynamical properties of transcriptional networks, but viewed from the angle of the application; in this case, a scalable wireless networking system. Here, nodes with communication capacity may continually enter or leave the system, which has parallels in molecular biology: proteins and other signaling biomolecules are continually made and destroyed, leading to uncertainty in the channel capacity of a signaling pathway. Our approach to this problem is to combining discrete event simulation and support vector machine learning techniques to identify important system features that contribute to the information flow across such networks. Discrete event simulation can capture dynamic behavior of the system by modeling information transmission as a set of independent events under custom perturbations using channel noise and congestion-based information loss; machine learning techniques can be used to identify underlying patterns in the data.

BICT 2014, December 01-03, Boston, United States
Copyright © 2015 ICST
DOI 10.4108/icst.bict.2014.257928

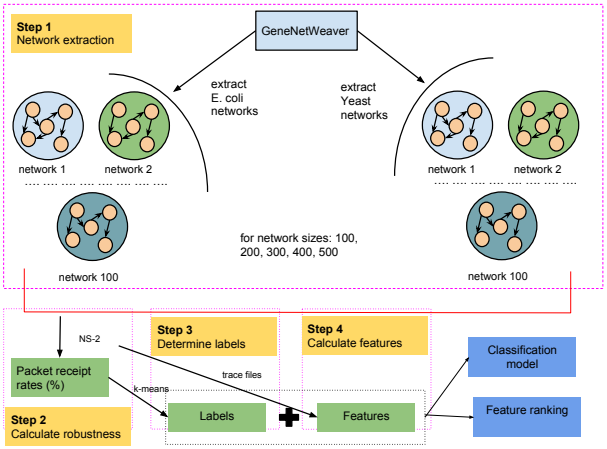


Figure 1: Illustration of the procedure followed in this work.

The NS-2 framework simulates information flow across wireless man-made systems in terms of packet transport, and we employ it here to quantify a type of dynamical network robustness by measuring the packet receipt rates at various destination nodes in the model networks. Packet receipt rate is determined as the ratio of number of packets successfully received at sink/destination nodes to the number of packets sent by the source node(s). While biological systems do not strictly communicate using information packets, they do employ signal transduction pathways that can be thought of a series of activation steps or 'checks,' which succeed upon passing a concentration threshold. This analogy can be taken further, given that biology is often redundant, in the sense that many pathways may be activated to achieve a single goal, reminiscent of flooding. We have described such similarities in detail before [3, 7, 8].

The results reported here build upon our previous work to explore properties crucial for robustness in transcriptional networks to design specialized wireless sensor network topologies [3, 7, 8], and quantifying performance of such networks using the NS-2 simulation framework [9].

2. METHODS

2.1 Model transcriptional networks

The GeneNetWeaver software package [16] is used here to extract subnetworks from transcriptional network datasets for the bacterium *Escherichia coli* and the common baker's yeast *Saccharomyces cerevisiae*. One hundred networks of five different network sizes $n = 100$, 200, 300, 400, and 500, as represented by the number of nodes n. For simplicity, we will refer to networks derived from *S. cerevisiae* as 'Yeast' networks, whereas the bacterial networks will be referred to as *E. coli* networks. For our purposes, we map the transcription factors as nodes, and transcriptional network edges represent are understood to denote interactions between participating nodes; thus, we ignored the regulatory interaction of each link. As a result, we may apply the concepts of graph theory [2] to the resulting networks.

2.2 Simulation setup

Network simulator (NS-2) software [13] is used here to simulate packet transmissions in the mapped network. Nodes corresponding to genes that code for transcription factors in the genetic network are taken as the source nodes, whereas nodes corresponding to nonregulating genes are considered to be the sink nodes. While source nodes can send and forward packets, sink nodes may only receive packets without forwarding them onto others.

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

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

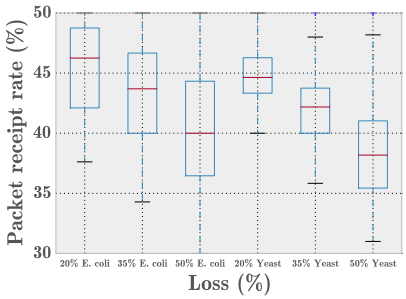
2.3 Motif structural redundancy and packet receipt

What is the the impact of structural redundancy, contributed by transcriptional motifs, on the information flow (packet transmission) through a complex network? In the context of the NS-2 framework, packets are successfully transmitted if those sent from a source node reach the sink (destination) node(s). That feed-forward loop transcriptional motifs (e.g. Fig. 3 (b)(1)) are hierarchical, and attenuate signal properties, such as response-time acceleration or delays [12], without any feedback loop, begs the question of whether they influence information transport at the more extensive network level. To examine this, we first tracked and identified all paths (node-hops) traveled by successfully received packets. We then used this history to identify all feed-forward loops that possess a nonempty intersection with these successful paths.

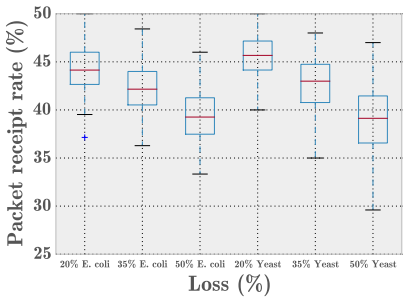
3. SUPPORT VECTOR MACHINE MODELING

Machine learning (ML) techniques can be used to discover and identify underlying patterns present in a given dataset. Currently, ML techniques are widely used for different purposes, such as to identify email spam, predicting election results, Internet search suggestions, targeted advertising, to name just a few. Among an army of techniques, support vector machine (SVM) is a supervised ML technique used for classification of data [4]. Our goal here is to first identify, and then to determine, which topological features of transcriptional networks best capture the behavior of a test network.

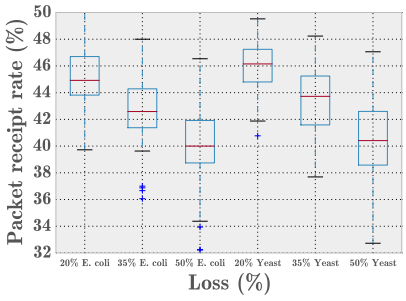
An SVM model identifies a *classifier* (boundary that separates data) which best classifies the given data. While linear classifier suits well in few instances, other instances may require non-linear separation boundaries. The implementation of such linear or non-linear boundaries in an SVM model is achieved using kernel functions. This classifier is often referred to as a hyperplane that separates instances



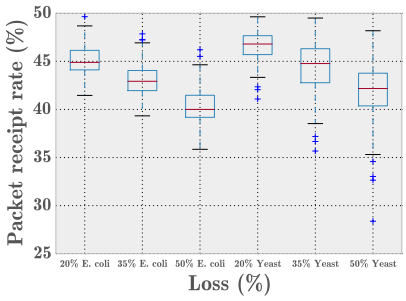
(a) $n = 100$ nodes.



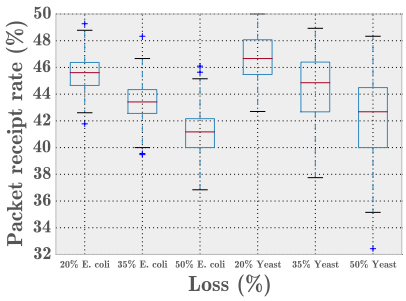
(b) $n = 200$ nodes.



(c) $n = 300$ nodes.



(d) $n = 400$ nodes.



(e) $n = 500$ nodes.

Figure 2: Packet receipt rates (PRTs) for sampled transcriptional subnetworks of the bacterium *Escherichia coli* and *Saccharomyces cerevisiae* (labeled 'Yeast').

belonging to different classes. The possible kernel functions include: linear, polynomial, radial basis function (RBF) and sigmoid. An SVM model predicts the target value of the test data given the features of test data.

An illustration of SVM dataset is shown in Figure 3(a). In SVM modeling, a dataset contains set of instances, and each instance is a combination of labels and features. The term 'label' is attributed to an output which describes a feature, which is a property of the dataset used. In addition, each feature is assigned a unique ID. For example, we employed ten datasets, which constitute five sampled subnetworks each from the transcriptional datasets for *Escherichia coli* and *Saccharomyces cerevisiae*. Each of these five datasets corresponds to a particular network size, as measured by the number of nodes, i.e. $n = 100, 200, 300, 400,$ or 500. One hundred networks were sampled from the source datasets for each size, and each such sampled subnetwork is an example of an 'instance'.

We used the Python programming language [18] and *scikit-learn* package [14] to identify features and build SVM classification models. *scikit-learn* utilizes the popular ML libraries *libsvm* and *liblinear*. We follow the data preprocessing and model selection steps as prescribed by [5]. We perform data scaling after feature determination (Section 3.5) then perform grid search (Section 3.4) to identify best parameters to classify data. Our goal is two-fold: a) to build a classification model b) rank features. The proposed classification model will be used in the future to predict new data. Feature ranking is performed using analysis of variance F-test which does not use model created by SVM.

3.1 Assigning labels for SVM

As shown in Figure 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 Figure 3. 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. 3(a)). The vectors X and Y together are termed as the dataset since it contains labels and features for a particular network size at a specific perturbation level.

3.2 Data pruning

A one-size-fits-all SVM model may not fully explain patterns within our datasets, such as statistical outliers of packet receipt from the NS-2 simulations, which become evident when clusters are identified using *k-means* clustering technique; because statistical outliers represent rare, large fluctuations, they may erroneously end up defining their own cluster. To avoid this problem, the dataset can be pruned by removing the labels and their corresponding data instances from the feature instances. Of course the best approach is to gather a maximum number of points to describe one network size, and this will be considered in future work. Consider the label vector Y with four clusters (IDs: $0, 1, 2, 3$)

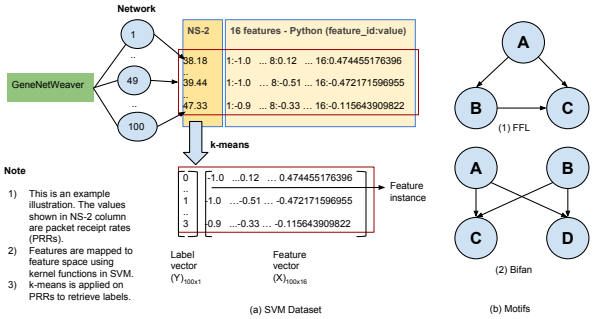


Figure 3: Illustration of (a) SVM Dataset for each network size, at specific perturbation level and (b)(1) FFL, (b)(2) bifan motifs respectively

to be $\{1:37, 0:34, 3:28, 2:1\}$. Only one point belongs to cluster ID 2 and hence that point is discarded along with the corresponding feature instance vector. Now, the training and testing is performed on Y which is 99 rows $\times 1$ column and X which is 99 rows $\times 16$ columns. In this work, data was not pruned.

3.3 Training and testing

Nevertheless, the pruned data is used as training and testing sets for the machine learning models. Each dataset is split into 75% training and 25% testing sets. In order to avoid overfitting the data, 5-fold cross validation is used to randomize the 75/25 split into training/testing datasets. In a 5-fold cross validation test, the split is performed five different times; labels are stored in a vector, and corresponding feature instances are stored in another, different vector. Continuing the example stated in the Section 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 can be understood as a weighted average of precision and recall [14]. The set of parameters used are shown in Table 1. C is the regularization constant and γ is a kernel hyperparameter [1] used in non-linear kernel functions. Large C overfits the data (high cost for misclassification). Large γ in polynomial kernel ensures a smoother decision boundary.

3.5 Features

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

In what follows, features defined based on the network topology are given in sections 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 referred to hereon as 'path-based features.'

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

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

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 information between them. For example, two adjacent nodes can be expected to communicate more frequently than two far-separated nodes in a noisy environment. We may compute this quantity according to the equation:

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

3.5.3 Degree centrality

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

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

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

3.5.4 Transcription factor percentage

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

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

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

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

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

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

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

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

3.5.7 FFL abundance

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

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

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

3.5.8 FFLDED

Figure 3(b)(1) illustrates a feed-forward loop transcriptional motif, which is hierarchical, but composed of two regulatory paths. The first is a 'direct' linkage from nodes A to C, whereas an 'indirect' path accounts for regulation of node C through a node B waypoint. Here, the feed-forward loop direct-edge density (FFLDED) is the ratio of feed-forward loop direct edges, $|E_{FFLDE}|$, to the total edges in the network, and may be calculated using the equation:

$$FFLDED = \frac{|E_{FFLDE}|}{|E|}. \tag{9}$$

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

3.5.9 FFLSSPD

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

3.5.10 FFLDEP

The FFLDED metric above (Eq. 9, accounts for the fraction of direct-edge feed-forward loop links present within the network topology. However, a single linkage may potentially appear more than once if it is 'shared' among two or more feed-forward loops. We define a separate measure that ignores multiple copies of any single link, which can be calculated as follows:

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

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

3.5.11 FFLIDEP

Indirect FFL edge percentage (FFLIDEP) is the ratio of the number of unique feed-forward loop indirect edges to the total number of sequential, two-step paths in the network. Thus, it is similar to the FFLDED metric above (Eq. 10), but measured against the indirect edge of the feed-forward loop. This can be calculated with the equation:

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

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

3.5.12 Direct-edge trace participation

Each NS-2 simulation results in a set of 'traces' that map packet-transport histories for packets sent and received successfully from source to sink nodes. In a similar concept to that of Eq. 9, but accounting for packet trace history, we measure the ratio of the number of unique feed-forward loop direct edges that participate in successful packet paths to the number of unique FFL direct edges, termed FFLDSPATH.

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

3.5.13 Indirect-edge trace participation

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

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

3.6 Feature ranking

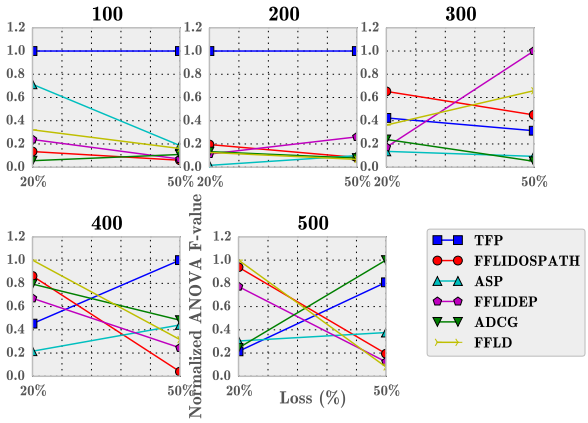
The identified features are ranked using the analysis of variance (ANOVA) F-value metric. This metric compares the inter-class variance to intra-class variance [14]. A higher F-value denotes higher significance of a feature. F-value captures feature significance individually but mutual feature dependence cannot be determined. We intend to use different metrics in the future work.

4. RESULTS

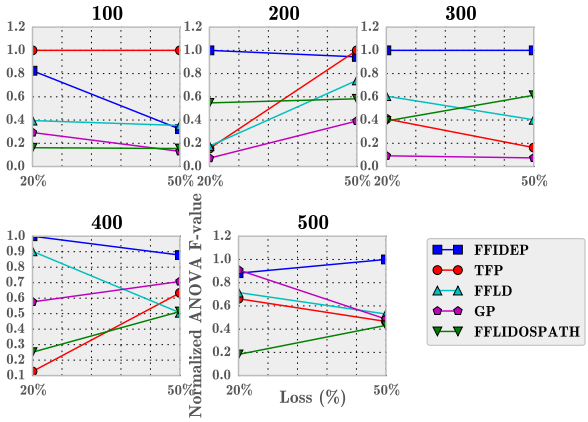
4.1 Packet receipt rates using transcriptional network topologies

Figure 2 illustrates the distribution of packet receipt rates (PRTs) for representative subnetworks sampled from *Escherichia coli* and *Saccharomyces cerevisiae*, across three different loss models (20%, 35% and 50%). Outliers in the dataset are points that do not occur in the range of top and bottom whiskers and are identified by +.

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

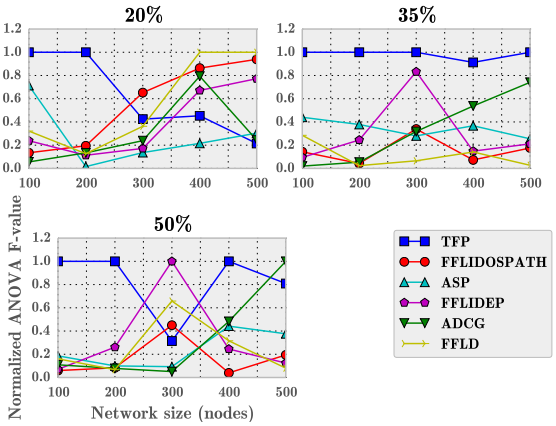


(a)

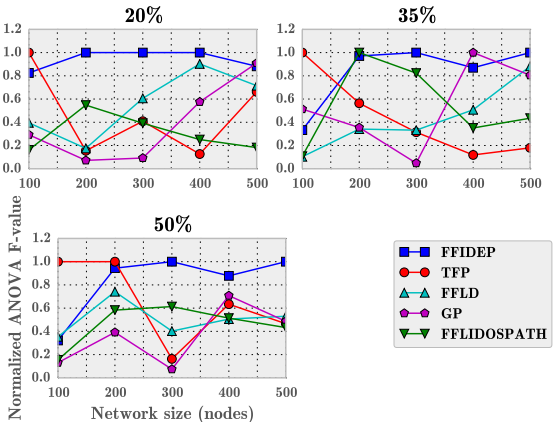


(b)

Figure 4: Variation of top 5 features in each *Escherichia coli* network (panel (a)) and *Saccharomyces cerevisiae* (panel (b)) networks, at losses 20% and 50% (Sizes = 100, 200, 300, 400, 500).



(a)



(b)

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

4.2 Feature ranking in transcriptional networks

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

4.2.1 Top-ranking features

Fifteen different SVM models, one for each pair of network size and perturbation level, are used to select features/metrics for one specific type of transcriptional network. Let us examine the feature selection in *E. coli* networks for one of the fifteen SVM model instances. For each network size, the top five features are selected, according to the criterion that each the most 'influential' features should occur at least three times in the top five features as scored across different network sizes. For *Escherichia coli* networks, this top-ranking set is given by the features: TFP, FFLIDOSPATH, ASP, FFLIDEP, ADCG, FFLD (Fig. 4a). Similarly, features so identified from the *Saccharomyces cerevisiae* networks are: FFLIDEP, TFP, FFLD, GP, FFLIDOSPATH (Fig. 4b). All influential features identified from the SVM models in terms of packet receipt rates relate to the feed-forward loop subnetworks.

4.2.2 Feature stability at different perturbation levels

As a preliminary experiment, we tested the prevalence of transcriptional network features at different noise perturbation levels. Here, our intention is observe if structural or dynamic features prevail in feature significance. The result of this on *E. coli* networks is shown in Figure 4a[2] and on Yeast networks is shown in Figure 4b. FFLIDEP ranks consistently higher in most cases (except at network size 100) than other features. Similarly, FFLD and GP rank in the top two or three at different network sizes. An interesting observation is that three (FFLIDEP, FFLD, FFLIDOSPATH) out of five top ranked features are related to FFL motifs.

4.2.3 Feature ranking variation across different network sizes

We now observe if the relative importance of features varied across different network sizes. From Figure 5a, it can be seen in *E. coli* networks that TFP ranks consistently stable in most cases in 35% and 50% perturbation levels (except at network size 300). FFLIDOSPATH, FFLD and FFLIDEP rank higher in some instances. Figure 5b shows the relative importance of features in Yeast networks. Here, FFLIDEP is relatively stable across different network sizes compared to other features. FFLD along with GP seems to be stable at certain instances but not conclusively overall. A combination of conventional metrics such as GP and motif-derived features can be used to engineer special networks which can ensure stability across different perturbation levels.

4.2.4 Comparison of FFL based features

Identifying features that are significant to network robustness will be substantial to design specially engineered networks that are *functionally* robust and can withstand perturbations. The results from the above two studies give us an opportunity to observe variation of FFL based features only instead of the top five identified features. A general trend can be observed from Figure 6 that FFL-based features have higher significance (based on normalized ANOVA F-value) from network sizes 300 and above. Second inference from Figure 6 is that FFLIDEP is ranked first among the six FFL based features in certain instances (100, 200, 300 and one instance in 400, 500 network sizes each). Figure 7 shows the ranking for Yeast networks. FFLIDEP ranks the highest for all network sizes and at all perturbation levels. Correlation between FFLDSPATH and FFLDOSPATH (derived from FFLDSPATH) is not always proportional suggesting that there is more to FFL participation and the number of successful FFL direct path contribution. The position of FFL might also be critical for prevalence of certain features. FFLDEP, FFLIDEP and FFLIDOSPATH consistently rank as the top three features at different perturbation levels. This directly reveals the importance of the percentage of FFL direct edges present in the network and the number of times those edges were used in successful packet transmissions.

[2]For the figure to be legible, X and Y labels are displayed only once. This is done for Figures 4a - 7.

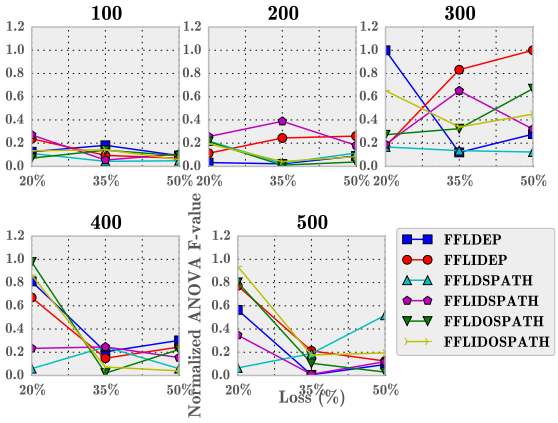


Figure 6: Variation of FFL participating direct and indirect edge-based features at 20%, 35% and 50% loss each for different E. coli networks (Sizes = 100, 200, 300, 400, 500).

5. DISCUSSION AND CONCLUSIONS

A key aspect before identifying and ranking features is mapping packet receipt rates to labels using *k-means* clustering algorithm. Choosing the optimal cluster size is crucial for creating labels. If one single point is equidistant from all different clusters, it will eventually remain in its own cluster. This problem can be addressed by gathering as many instances as possible for a given network size. Training to testing data set split ratio is critical for creating a classification model. Selecting a high training set percentage will overfit the data. Another challenging aspect is the data loss due to pruning (as explained in Section 3.2). Feature ranking can potentially be influenced by inappropriate data pruning. Using sufficient number of data instances can address this problem.

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

We studied transcriptional networks of the model bacterium *Escherichia coli* and the common baker's yeast *Saccharomyces cerevisiae* to identify system-defining features

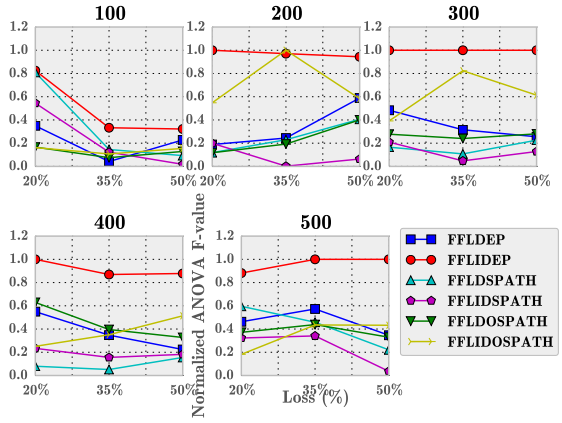


Figure 7: Variation of FFL participating direct and indirect edge-based features at 20%, 35% and 50% loss each for different Yeast networks (Sizes = 100, 200, 300, 400, 500).

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

6. ACKNOWLEDGEMENTS

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 authors thank Ljiljana Zigic for discussions on SVM classification and regression modeling.

7. REFERENCES

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

[2] J. A. Bondy and U. S. R. Murty. *Graph theory with applications*, volume 6. Macmillan London, 1976.

[3] 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.

[4] M. A. Hearst, S. Dumais, E. Osman, J. Platt, and B. Scholkopf. Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13(4):18–28, 1998.

[5] C.-W. Hsu, C.-C. Chang, C.-J. Lin, et al. A practical guide to support vector classification, 2003.

[6] M. Isalan, C. Lemerle, K. Michalodimitrakis, C. Horn, P. Beltrao, E. Raineri, M. Garriga-Canut, and L. Serrano. Evolvability and hierarchy in rewired bacterial gene networks. *Nature*, 452:840, 2008.

[7] 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.

[8] 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*, pages 1–17, 2014.

[9] B. K. Kamapantula, A. F. Abdelzaher, M. Mayo, E. J. Perkins, S. K. Das, and P. Ghosh. *Quantifying robustness of biological networks using NS-2*. Springer (Under revision), 2014.

[10] H. Kitano. Towards a theory of biological robustness. *Molecular systems biology*, 3(1), 2007.

[11] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási. Controllability of complex networks. *Nature*, 473(7346):167–173, 2011.

[12] 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.

[13] S. McCanne, S. Floyd, K. Fall, K. Varadhan, et al. Network simulator ns-2, 1997.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[15] 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.

[16] 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.

[17] S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nat. Genet.*, 31(1):64–68, 2002.

[18] G. Van Rossum et al. Python programming language. In *USENIX Annual Technical Conference*, 2007.