Feature ranking in transcriptional networks: Packet receipt as a dynamical metric

Bhanu K. Kamapantula*
Virginia Commonwealth University
401 W Main st
Richmond, VA, USA
kamapantulbk@vcu.edu

Michael Mayo
Environmental Laboratory, US Army Engineer Research and Development Center
Vicksburg, MS 39180
Michael.L.Mayo@usace.army.mil

Edward Perkins
Environmental Laboratory, US Army Engineer Research and Development Center
Vicksburg, MS 39180
Edward.J.Perkins@usace.army.mil

Ahmed F. Abdelzaher†
Virginia Commonwealth University
401 W Main st
Richmond, VA, USA
abdelzaheraf@vcu.edu

Preetam Ghosh
Virginia Commonwealth University
401 W Main St
Richmond, VA, USA
pghosh@vcu.edu

ABSTRACT

Machine learning techniques may be useful in determining the features contributing to some biological properties, such as robustness, which is the tendency for biological systems to resist a change of state. In this work, we compare transcriptional subnetworks extracted from the bacterium *Escherichia coli* and the baker's yeast *Saccharomyces cerevisiae* using *in silico* experiments. We use the packet receipt rate as a metric to quantify biological robustness, which is different from the usual structural metrics since it captures the dynamic behavior of the network. We define seventeen features based on structural significance, such as transcriptional motifs, and conventional metrics, such as average shortest path and network density, among others. Feature ranking is performed, based on a grid-search method to identify Support Vector Machine classifier parameters using cross validation. Our results indicate that feed-forward loop based features are important for bacterial transcriptional networks, whereas network density, degree-centrality based and bifan-based features are found to be significant for yeast-derived transcriptional networks. Interestingly, results suggest that feature significance varies with network size (number of nodes). As a first, this study quantifies the impact of the feed-forward loop and bifan transcriptional motif abundance observed in natural networks.

*Corresponding author - kamapantulbk@vcu.edu
†abdelzaheraf@vcu.edu

BICT 2014, December 01-03, Boston, United States

DOI 10.4108/icst.bict.2014.257930

General Terms

Machine learning, feature ranking, complex networks, transcriptional networks

Keywords

biological robustness, transcriptional network

1. INTRODUCTION

We investigate genetic regulatory networks (GRNs) obtained from the bacterium *Escherichia coli* (*E. coli*) and the common baker's yeast *Saccharomyces cerevisiae*, referred to herein as Yeast, with the aim of uncovering the primary contributions to robustness exhibited by these networks. Such genetic networks are known to be robust in function, despite exposure to disruptions such as genetic manipulation [10]. In some cases, robustness can be attributed to properties of the network structure, such as with a tolerance to mutations originating with the power-law degree distribution [1] or to dynamic stability correlated with the abundance of specific transcriptional motifs, such as the feed-forward loop [19]. The feed-forward loop is just one example of a subnetwork found more abundantly in nature than in random networks, collectively network motifs [21]. Special functionality has been attributed to the feed-forward loop, such as signal-pulse generation and changing response times [17]. In this study on significant network features, we explore the performance of Support Vector Machine (SVM) features derived from transcriptional subnetwork motifs, such as the feed-forward loop or bifan motif (a feed-forward loop motif ABC consists of edges AB, BC, AC and a bifan motif ABCD consists of edges AC, AD, BC, BD. All edges are directed). An illustration of FFL and bifan is shown in Figure 1.

Here, we hypothesize that significance of structure-derived features of transcriptional networks can be explored using SVM methods. To test this, we first map the gene-gene and transcription factor-gene interactions obtained from organism-based GRNs to a network of nodes and edges, wherein genes and transcription factors are represented by nodes and the

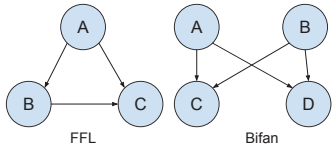


Figure 1: Illustration of FFL and bifan motifs.

edges represent the interactions between interacting nodes. Once the biological system is mapped into a directed network, the principles of graph theory [2] may be applied to study its structural characteristics. Part of these methods include control theoretical ideas, which can be employed to alter properties of critical network entities (e.g., nodes or communities), or to manipulate larger portions of a network [16]. Overall, an improved understanding of the primary topological features which contribute to certain biological properties, such as robustness, may lead to new strategies and algorithms for building randomized network models prescribed, *ab initio*, with these or other desired properties.

This work is built upon previous research that aimed to understand aspects of robustness present in transcriptional networks, and exploring how to engineer these properties into wireless sensor network topologies [11, 12, 6], quantifying performance of the test networks using NS-2 [13]. As such, this paper is organized as follows. Section 2 provides a discussion on how robustness is described in the literature across various network contexts. Section 3 details the extraction of networks, the simulation setup, and the procedure used to assess results. Section 5 describes the SVM model and feature ranking method employed in the model subnetworks of the biological ones.

2. BACKGROUND

2.1 Measures of biological robustness

Definitions of biological robustness are varied, and several have been proposed in the last decade. Robustness is often defined as an ability of a system to withstand some level of disruption, and to perform tasks as intended. Kitano's version embodies this spirit, wherein he described robustness as "...a property that allows a system to maintain its functions against internal and external perturbations" [14]. However, robustness is not a general property *per se*, but relies on specific context, and is usually expressed as a property of a particular metric. For example, dynamical robustness might refer to persistence of a steady-state, despite small dynamical perturbations [19].

In the realm of complex networks and control theory, robustness of controllability is defined as the ability to control aspects of the whole network through manipulation of a smaller subset of nodes [16]. In this spirit, several node-based metrics have been proposed to quantify robustness in complex networks; some are centrality-based (e.g., degree, betweenness, or closeness centrality) and path-based (e.g.., average shortest path, communicability). Network "strength" has also been proposed, based on the relative interactions level of the connected components [1].

A recent study by Chan *et al.* [3] explored the idea of robustness in complex networks, and one metric they used was derived from the Estrada index [5], which is only applicable to undirected networks. Moreover, this is a structural metric which cannot uniquely capture the dynamic behavior of the transcriptional system, given that such links additionally modify the regulatory state of the target node. We should note that none of their analyzed metrics considered impacts from motif abundance, which has been previously correlated with dynamical robustness [19]. One network-based metric that captures the dynamical aspects of information flow over a network is the packet receipt rate (PRR), which we define here as the ratio of the number of information packets received at one or more given sink nodes to the number of packets sent by the source nodes.

2.2 Packet receipt as a dynamical metric

We use the network simulator NS-2 as a simulation framework in which to quantify the packet receipt rate. However, an explanation of its applicability to transcriptional network is in order, because biological systems don't obviously communicate via information packets, but rather pathway transduction facilitated by passing a sequence of concentration thresholds. We have previously described a suitable mapping for these purposes [6, 11, 12], and briefly explain it below.

As explained above, a gene regulatory network can be reduced to its constituent network of nodes and links, wherein the transcription factors (TFs) and genes are nodes and the interactions among them are represented by edges. This basic structure may be used to transmit information packets from source nodes, which may be received at sink nodes (genes). This scheme necessitates selection of suitable source and sink nodes, and we have explored several methods before [6, 11]. We restrict transmission properties of the network, such that genes may only receive packets, but TFs may both send and forward packets; our sink-selection method is one wherein all nodes with zero out-degree are labeled as sink nodes, which may only receive packets. Finally, we express the packet receipt rate in terms of parameters specific to the communication network, such as network loss, packet transmission rates, and queue limits.

3. METHODS

Figure 2 provides a graphical depiction of the procedure followed in this work. Section 3.1 describes the network extraction–as illustrated in Figure 2 (Step 1)–from *E. coli* and Yeast model networks using the GeneNetWeaver software [20]. Section 3.2 details the simulation setup and the determination of robustness–illustrated in Figure 2 (Step 2)–using NS-2 software. Finally, sections 5.1 and 5.2 discuss label determination using k-means clustering algorithm and feature calculation.

3.1 Bacterial and yeast derived networks

Networks of five different sizes were used, measured in terms of the total number of nodes: 100, 200, 300, 400 and 500. For each size, we extracted 100 sample networks from either *E. coli* or *S. cerevisiae* transcriptional networks using GeneNetWeaver. Herein we refer to the model networks derived from either *E. coli* or *S. cerevisiae* as, respectively, *E. coli* or Yeast networks.

In the analyses explained below, two networks were compared if they were of equal size (i.e. number of nodes), although individual networks may support a variable number of total links. For two such subnetworks, labeled A and B, we hypothesize that if the networks exhibit robustness in the

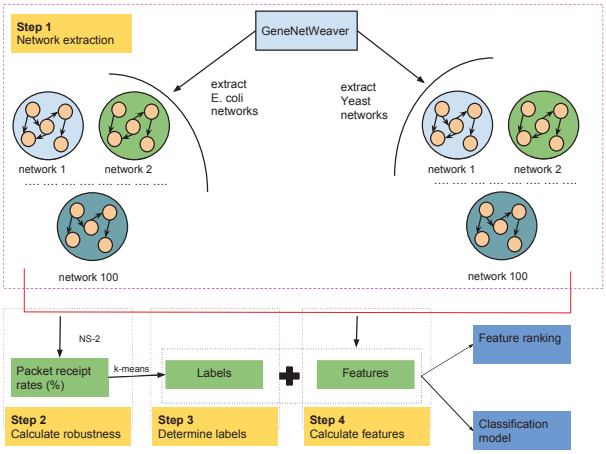


Figure 2: Illustration of the procedure followed in this work.

dynamical metric (the packet receipt rate), then there exists identifiable topological features attributable to the variation between subnetwork A and B. To test this hypothesis, we carried out *in-silico* simulations to identify the best, average and worst performing networks, as ordered by the numerical value of their packet receipt rate(s).

Packet receipt rates were measured for each network against varying levels of "loss" in a given communication channel. Higher levels of loss correlate with a greater chance that transmitted packets fail to arrive at the destination node.

3.2 NS2 simulation setup

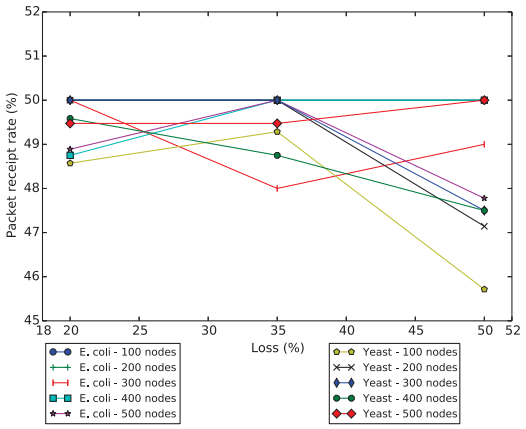
This work uses our previous approach to quantify network robustness using packet receipt rate in [11, 12, 13]. Network robustness is measured across three different loss models: 20%, 35% and 50%. Queue limit at a node is set at five (packets). All edges are considered to be directed. Nodes with zero out-degree are considered to be sinks and other nodes are considered to be source nodes. While sink nodes only receive packets, source nodes transmit and forward the packets. This scenario resembles a biological system where transcription factor(s) regulate gene(s). Packet transmission follows flooding protocol wherein each node can send ten packets to its direct neighboring nodes. Packet receipt rate is calculated as the ratio of number of packets received at all sinks to the number of packets generated at source nodes. We represent robustness of a network as a percentage *(packet receipt rate)*$*$100.

Since the simulations consider channel fluctuation and congestion based packet drops as perturbations, higher robustness value for a network makes it *more* robust compared to a network with lower robustness value.

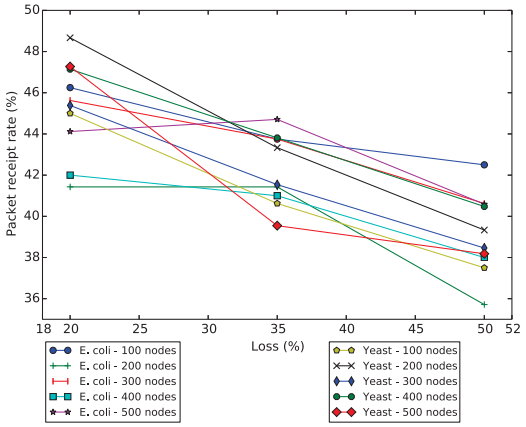
4. NS-2 SIMULATION RESULTS

Packet receipt rates of best, average and worst performing networks for sizes 100, 200, 300, 400 and 500 are presented in Figures 3a, 3b, and 3c.

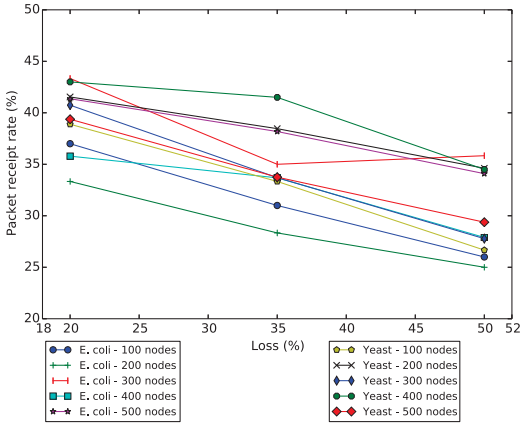
To compare network performance, we estimated the area under the plotted curves illustrating the packet receipt rate to the channel loss (Figs. 3(a-c)). Linear interpolation was used to extrapolate PRR trends between networks of varying size, and the trapezoidal rule was used to calculate the area under the curve. Out of fifteen instances (five differ-



(a)



(b)



(c)

Figure 3: Comparison of (a) best, (b) average and (c) minimum performing *E. coli* and Yeast derived networks, respectively, for sizes 100, 200, 300, 400, and 500 nodes

ent network sizes and three different loss models) eleven *E. coli* derived networks performed "better" than their counterparts, in the sense of higher PRR levels despite conditions of channel loss. In three cases (200, 400, 500 sizes), Yeast derived networks performed better than their counterparts. In one case (100 network size, Fig. 3(a)), both performed identically. In certain instances (*E. coli* network - 300 nodes at 35% and 50%), higher fluctuation in loss might not result in drop in PRR since there are multiple options for a packet to be transmitted to sink(s). Full impact of the effect of multiple sinks across different perturbations needs to be explored.

Selective features can contribute for better performance of a specific network. For example, the reasons for *robust* behavior of a network can be due to structural redundancy. Motifs such as feed-forward loop support structural redundancy that can be captured using NS-2 framework. For one such motif ABC (with directed edges AB, AC and BC), information can be transmitted from A to C directly or via B. In order to identify these features, we use Support Vector Machine (SVM) methods where network data is trained and tested using a classifier. This is described below in Section 5. We define features related to feed-forward loop motifs among others to understand such behavior. Using the SVM classifier, we rank features. We first measured a set of features identified as important by earlier results from the literature, and then created additional features to generate a more holistic picture. The next section defines SVM concepts and describes the features and the methodology used to determine significant features among them.

5. SUPPORT VECTOR MACHINE MODELING

Machine learning (ML) is now widely used by businesses to identify email spam, predict airline prices on a busy weekend, predict football game outcomes, predict national election outcomes, credit card fraud detection among a slew of other applications. For example, one application involves development of a feature detector to identify Human bodies and cat faces, using unsupervised learning techniques [15]. As more data regarding cellular interactions become available, such techniques can be employed to develop improved predictive models of cellular regulation. In the context of biological systems, [8] used k nearest neighbors classifier to predict cellular localization sites of proteins in *E. coli* and yeast. ML algorithms such as decision tree, bayes prediction, logistic regression are used by [4] to predict metabolic pathways. Unsupervised ML algorithms are used to predict the output (labels) of new datasets based on models built using historical data. Supervised learning methods are used to study and classify labeled data. Labels can be understood as output for a set of described data properties. SVMs, a type of supervised learning technique, are used here to identify significant patterns in *E. coli* and yeast networks. In our case, we use packet receipt rates as labels (after mapping them: Section 5.1) and features (Section 5.2) as properties that describe the network data. Features in our case are structural properties and properties derived from transcriptional motifs. Each network instance (hundred instances) for each network size (five sizes) is labeled with an output which is packet receipt rate calculated using NS-2 simulations. Table 1 describes the data format used to build SVM classifier for a given network size.

Table 1: Data format used to build SVM classifier for each network size and specific network type (*net* refers to network and *f_id*, *f_value* refer to feature id and value respectively)

SVM input data format
net1_output f_id1:net1_f_value ... f_id17:net1_f_value
....
net100_output f_id1:net100_f_value ... f_id17:net100_f_value

The network datasets obtained above through the GeneNetWeaver software toolkit were processed using SVM classification models [7]. We follow the data preprocessing and model selection style defined by [9]. The features were scaled to the range $[-1, 1]$ to avoid artificial bias toward high-valued features, and we applied this scaling procedure to all network datasets.

5.1 Label mapping using k-means algorithm

Robustness values of networks fall in the range of $0.0 - 100.0$. In order to build a SVM classification model, integer labels are required. Hence, we map the robustness labels (in floating point) to integer values. This is performed using k-means clustering algorithm instead of arbitrary allocation. For a given set of n points, k-means algorithm partitions the points into k clusters. Initially, the points are clustered with a random center for each cluster. Then, the distance of each point to all the cluster centers is estimated and the point is reassigned to the cluster center nearest to it. This process is continued until the centers no longer change. For this work, we grouped the data into five clusters. Now, this data is used to perform grid search and identify best parameters to build an SVM classifier.

5.2 Features

Feature extraction is a critical aspect before choosing the ML model. We define certain features based on accepted metrics in research community. Average shortest path measures the capability of any two nodes in the network to communicate and hence is chosen to be explored. Network density captures the sparsity of nodes in the network. We also measure centrality metrics such as degree centrality, betweenness centrality and closeness centrality as they identify nodes that work as hub nodes for information flow in a network. Feed-forward loop (FFL) motif has been identified to contribute to robustness–preserving system function despite internal and external perturbances–in genetic networks [14]. FFLs are also have been shown to be important for biological functions such as generating signal pulses, and speeding up or delaying response times [17]. Hence, three FFL-based metrics are defined as features. FFL motifs, despite being responsible for several biological functions are found to be less stable than bifan motif [19]. Hence, three bifan-based metrics are defined as features.

A total of seventeen features are considered to build the SVM classification model. We define each feature before we identifying the significant ones.

5.2.1 Network density

Network density (ND) is the amount of edges present in the network compared to the total number of edges possible

in the network [1].

5.2.2 Average shortest path

Average shortest path of the network (ASP) is the ratio of the sum of shortest paths for all pairs of nodes to the total number of possible edges.

5.2.3 Genes percentage

Genes percentage (GP) is the percentage of gene nodes with respect to the total nodes in the network.

5.2.4 Transcription factors percentage

Transcription factor percentage (TFP) measures the number of transcription factor nodes compared to the total number of nodes in the network.

5.2.5 Transcription factor network density

Transcription factor network density (TFND) determines the percentage of total edges connected to transcription factor nodes.

5.2.6 Genes coverage

Genes coverage (GC) is the ratio of in-degree of all sink nodes to the sum of the number of source nodes with paths to the sink nodes.

5.2.7 Centrality measures

a) Degree centrality

For a node, degree centrality determines the fraction of nodes it is connected to in the network. The average degree centrality of gene nodes (ADCG) and transcription factor nodes (ADTF) are considered separately. Simultaneously, average degree centrality of the network (ADC) is considered.

b) Betweenness centrality

Betweenness centrality of a node is the number of shortest paths the node participates in compared to the total number of shortest paths in the network. The average betweenness centrality of transcription factor nodes (ABTF) is considered.

c) Closeness centrality

Closeness centrality measures the distance of each node to all other nodes. Here, the average closeness centrality of transcription factor nodes (ACCTF) is considered. A normalized version of the closeness centrality is used as a feature.

It can be observed that betweenness centrality and closeness centrality for gene nodes are not considered as they do not participate as intermediate nodes in shortest paths since their out-degree is zero. Eigenvector centrality metric is not considered since the convergence using power method is not possible for all the networks.

5.2.8 FFL edge abundance

FFL edge abundance (FFLD) measures the number of edges participating in FFLs compared to the total number of edges in the network.

5.2.9 Bifan edge abundance

Bifan edge abundance (BFD) measures the number of edges participating in bifans compared to the total number of edges in the network.

[1]Equations describing the feature determination for all features have been removed for space considerations.

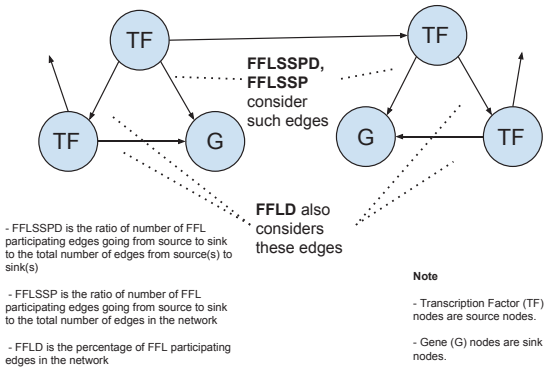


Figure 4: Demonstration of FFLSSP, FFLD and FFLSSPD features

5.2.10 FFLSSP

We determine the total edges such that each edge participates in an FFL and also goes from a transcription factor node (source) to a gene node (sink). Two features are derived from this metric: a) the count determined is compared to the total number of edges in the network (FFLSSP) and b) the count determined is compared to the total number of direct edges from transcription factor nodes to gene nodes (FFLSSPD). Figure 4 illustrates this scenario.

5.2.11 BifanSSP

We determine the total edges such that each edge participates in an bifan and also goes from a transcription factor (source) node to a gene node (sink). Two features are derived from this metric: a) the count determined is compared to the total number of edges in the network (BFSSP) and b) the count determined is compared to the total number of direct edges from transcription factor nodes to gene nodes (BFSSPD). Figure 4 can be extended to bifan motif.

All features are scaled from -1 to 1 based on the Equation 1.

$$F_{js} = (\frac{F_j - F_{min}}{F_{max} - F_{min}}) * 2 - 1 \quad (1)$$

where 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 the F.

5.3 Implementation and feature ranking

Python programming language [22] is used to implement the feature ranking and SVM classification model. We used *scikit-learn* [18] package developed using Python for SVM classification feature ranking and building a classification model. *scikit-learn* uses the popular *libsvm* and *liblinear* packages internally.

As recommended in [9], the best classifier parameters are determined using grid search with ten-fold cross validation. Cross validation is performed to avoid overfitting the data. We considered linear, RBF and polynomial kernels as kernel options [2]. Initially, the classifier was modeled for ten-fold

[2]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

Table 2: Best grid search parameters using cross validation - *E. coli*, yeast-derived networks

Network size(s)	Kernel	C	Gamma (γ)	degree
yeast - 100, 300	Polynomial	1, 1	1, 1	3, 3
yeast - 200, 400, 500	RBF	10, 1, 10	1, 1, 2	-
E. coli - 100, 200, 400, 500	RBF	100, 10, 1, 100	0.1, 2, 1, 0.1	-
E. coli - 300	Polynomial	1	1	4

cross validation. In some instances, number of labels for a class was found to be less than the number of folds (ten). Hence, five-fold cross validation is performed. Best parameters are identified by taking the mean of accuracy across five-fold cross validation. For this study, training data is set at 85% of the entire data and remaining data is for testing purposes. Hence, 85% of the data is used for training purposes [3]. Here, the training set is divided into ten sub-datasets of equal size and each sub-dataset is tested using the classifier trained on the remaining nine sub-datasets. This is done for each C & γ pair. Once the best parameters (defined in Tables 2 for both yeast and *E. coli* networks) are identified, feature ranking and classification model building is performed. An SVM classification model is built for future purposes to predict the performance of extracted subnetworks. Accuracy score[4] is used to identify the accuracy of the classifier.

Features are ranked using analysis of variance (ANOVA) F-value metric. ANOVA F-value calculates the ratio of inter-class variance to within-class variance. This metric is used from scikit-learn [18]. A higher F-value denotes higher significance of a feature. In this work, we filter top five features out of the defined seventeen features. While F-value calculates the feature significance individually, mutual feature dependence cannot be estimated by this metric. We intend to address this aspect in the future work.

5.4 Feature significance

For each network size and specific network type, an SVM model is created and corresponding features are ranked. Considering yeast networks first, nine important features are plotted in Figure 5. Top five features are ranked for each network size. The superset of all features for five different network sizes is then selected for comparison. ND and ADC rank higher than most features for yeast networks. BFD and BFSSP also feature among the top ranked features. Similarly, this is repeated for *E. coli* networks. For *E. coli* networks, from Figure 6 it can be observed that there is no feature that is a clear winner. An interesting observation is the change in feature ranking for change in network size. For instance, BFD ranks higher for network sizes 300 and 400 and ADTF ranks higher for network sizes 200, 300 and 400.

5.4.1 *Normalized features*

To understand the relative importance of features, normalized ANOVA F-values of all the features is determined and illustrated in Figures 7a, 7b, 7c, 7d, 8 and 9. At any given network size, more number of features rank higher for *E. coli* networks than yeast networks. A combination of the highly ranked features can be used to create specialized networks in the future which can ensure maximum efficiency.

5.4.2 *Feature comparison in* E. coli *and yeast networks*

It can be observed from Figs. (7a-7d, 8) that FFLSSPD, FFLSSP and FFLD rank higher for E. coli networks than yeast networks. In biological context, this information is crucial. Since FFLSSPD and FFLSSP consider both the number of edges in FFLs and direct edges from sources to sinks, the essence of network robustness is captured in these metrics.

5.4.3 *Feature trend*

Figure 9 is used to observe the trend of individual features. TFP, GP, BFD, BFSSP, ND, ABCTF, ACCTF rank relatively higher across all cases in *E. coli* networks than in yeast networks. ADTF ranks higher in yeast networks than its counterpart. ADC performs better in yeast networks at few network instances (200, 400) than *E. coli* networks. This study will help design flexible learning classifiers where features can be adaptively used as plug-ins depending on the network type. Since, bifan-based features work better for yeast networks, bifan interaction within these networks can offer insights for adaptive information transmission in a network.

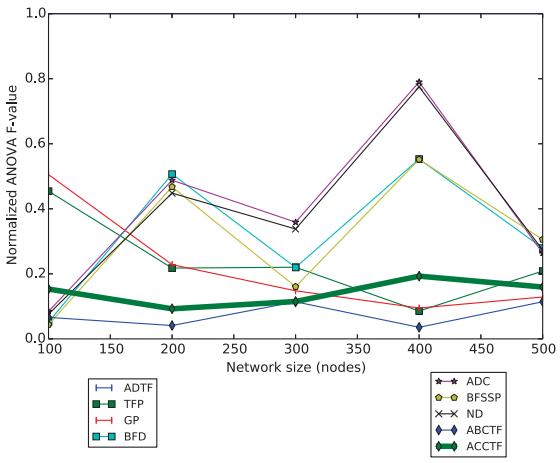


Figure 5: Comparison of 8 features across Yeast networks of sizes 100, 200, 300, 400 and 500.

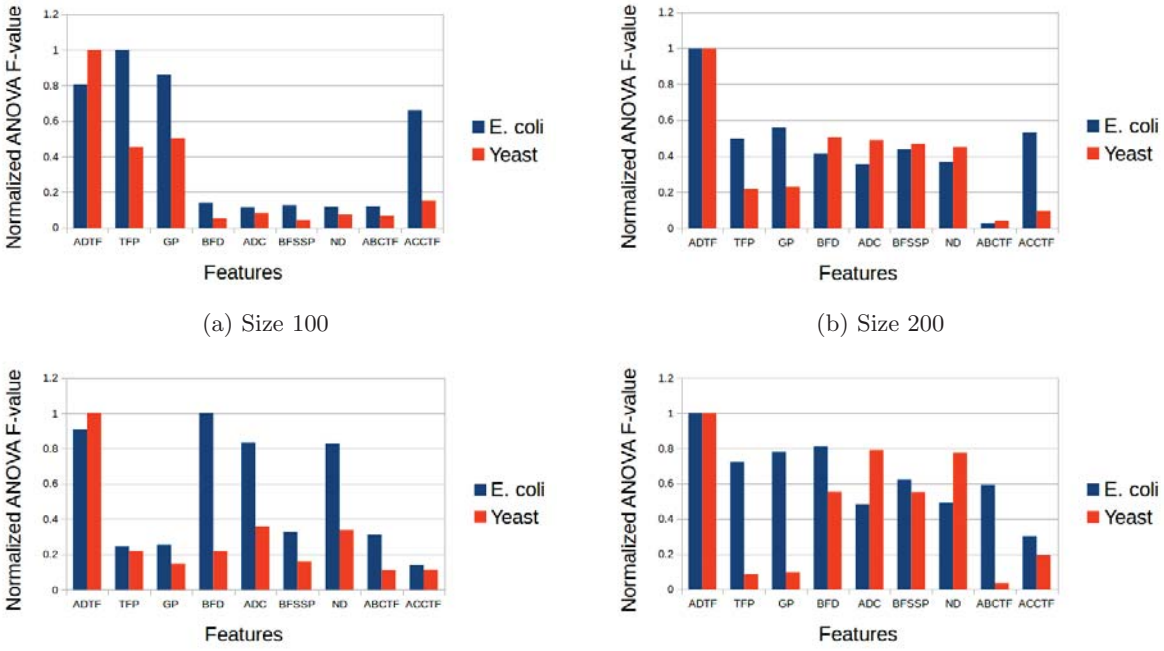
6. DISCUSSION AND CONCLUSIONS

To compare the relative efficiency of *E. coli* and yeast networks, we use quantitative methods to simulate packet transmission in the subnetworks derived from their regulatory networks. As a first, we identify several features that

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).

[3]Data were also modeled by using 75%:15% data ratio for training and testing. No significant difference was noticed in estimating the features.

[4]It is the ratio of number of true predicted values to the true values.



(a) Size 100

(b) Size 200

(c) Size 300

(d) Size 400

Figure 7: Feature comparison *E. coli* and Yeast networks for sizes 100, 200, 300 and 400.

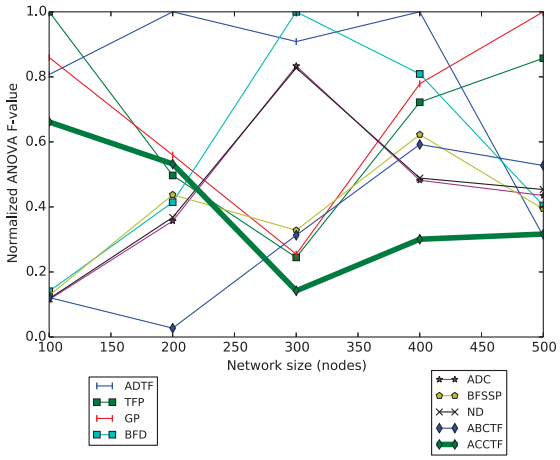


Figure 6: Comparison of 8 features across *E. coli* networks of sizes 100, 200, 300, 400 and 500.

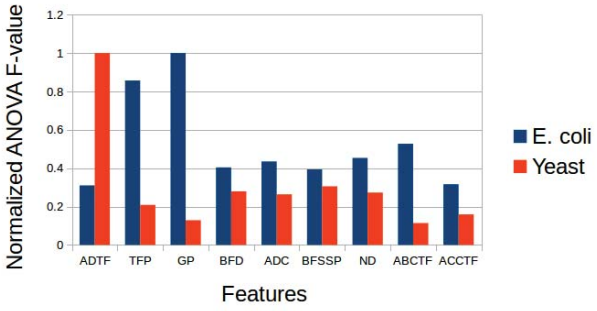


Figure 8: Feature comparison *E. coli* and Yeast networks for size 500.

potentially contribute to the robustness of networks derived from both organisms. Feature ranking is performed to identify significant features using normalized ANOVA F-value and the superset of top five identified features, across different network sizes, is determined. While ADTF ranks distinctly higher than other features for yeast networks, [ND, ADCG, ADC, BFSSP, BFSSPD, BFD] mostly outrank other features for *E. coli* networks. To the best of our knowledge, this is the first study to compare *E. coli* and yeast networks using feature ranking.

Machine learning can be a critical tool to understand biological principles. Our classifier will be improved in the future by pruning insignificant features. Extensive study using larger sample size will be carried out to understand the significance of label mapping using k-means clustering, choice of training and testing data split ratio. This work

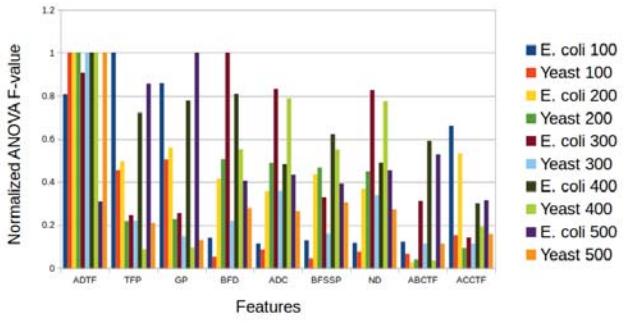


Figure 9: Normalized ANOVA F-value of features calculated for a specific network type and size for *E. coli* and Yeast networks.

paves a new way to compare biological systems and design bio-inspired topologies. Specialized bio-inspired networks can be designed to exploit the identified features such as ND, ADCG, ADC and biological features such as FFLSSP, FFLSSPD, FFLD, BFSSP, BFSSPD, BFD that are derived based on functionally important FFL and bifan motifs. Selective feature usage will help maximize information transmission and help realize network efficiency.

7. ACKNOWLEDGEMENTS

This work was partially funded in part 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.

8. 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] 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.

[4] J. M. Dale, L. Popescu, and P. D. Karp. Machine learning methods for metabolic pathway prediction. *BMC bioinformatics*, 11(1):15, 2010.

[5] J. A. de la Peña, I. Gutman, and J. Rada. Estimating the estrada index. *Linear Algebra and its Applications*, 427(1):70–76, 2007.

[6] 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.

[7] 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.

[8] P. Horton and K. Nakai. Better prediction of protein cellular localization sites with the it k nearest neighbors classifier. In *Ismb*, volume 5, pages 147–152, 1997.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

[15] Q. V. Le. Building high-level features using large scale unsupervised learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8595–8598. IEEE, 2013.

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

[17] 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.

[18] 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.

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

[20] 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.

[21] 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.

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