International Journal of Scientific Research & Engineering Trends
Volume 5, Issue 2, Mar-Apr-2019, ISSN (Online): 2395-566X

Comparative Study on Microarray Data Analysis Techniques of Machine Learning

M.Tech. Scholar Sanjay Kumar Sharma
Dept of CSE,
MITS B hopa l, MP, India
sssks6913@gmail.com

Asst. Prof. S.R. Yadav
Dept. of CSE
MITS Bhopal, MP, India
techmillenniumk.yadav@gmail.com

Abstract- **Machine learning models has been applied for prediction and prognosis of disease development and it become an irrevocable part of microarray data analysis aimed at betterment of the subsequent therapy and management of patients. The application of machine learning models for accurate prediction of survival time in cancer disease on the basis of clinical data is research work of the presented study. The paper discusses an approach to the problem in which the main factor used to predict survival time is the originally developed tumor-integrated clinical feature, which combines tumor stage, tumor size, and age at diagnosis. Two datasets from corresponding liver cancer studies are united by applying a data integration approach based on horizontal and vertical integration by using proper document-oriented and graph databases which show good performance and no data losses. Aside from data normalization and classification, the applied machine learning methods provide promising results in terms of accuracy of survival time prediction. Clustering has been used extensively as a vital tool of data mining. A neuro-fuzzy method is proposed in this research for analyzing the gene expression data from microarray experiments. Analysis of gene expression data leads to cancer identification and classification, which will facilitate proper treatment selection and drug development. The proposed approach was tested on three benchmark cancer gene expression data sets. Experimental results show that neuro-fuzzy method can be used as an efficient computational tool for microarray data analysis. The neuro-fuzzy classification system, which is based on a built clustering algorithm reached recognition rates than other classifiers. Gene expression data sets for liver cancer were analyzed in this research. A training and test data set for each cancer was used to analyze the quality of the genes. Use of computational methods to predict Gene Regulatory Networks (GRNs) from gene expression data is a challenging task. Many studies have been conducted using unsupervised methods to fulfill the task; however, such methods usually yield low prediction accuracies due to the lack of training data. In this research work, semi-supervised methods have been proposed for GRN prediction by utilizing two machine learning algorithms, namely support vector machines (SVM) and random forests (RF). The semi-supervised methods make use of unlabeled data for training. Inductive and transductive learning approaches has been investigated, both of which adopt an iterative procedure to obtain reliable negative training data from the unlabeled data. Then semi-supervised methods are applied to gene expression data of Escherichia coli and Saccharomyces cerevisiae, and evaluated the performance of the methods using the gene expression data. This analysis indicated that the transductive learning approach outperformed the inductive learning approach for both organisms. However, there was no conclusive difference identified in the performance of SVM and RF. Experimental results also showed that the proposed semi-supervised methods performed better than existing supervised methods for both organisms.**

Keywords- **Machine learning, Liver cancer, Neuro-Fuzzy survival time prognosis; Normalization, Classification, Clustering.**

I. INTRODUCTION

1. Background

A **gene** (or **genetic**) **regulatory network** (**GRN**) is a collection of molecular regulators that interact with each other and with other substances in the cell to govern the gene expression levels of mRNA and proteins. These play a central role in morphogenesis, the creation of body structures, which in turn is central to evolutionary developmental biology. The regulator can be DNA, RNA, protein and complexes of these. The interaction can be direct or indirect (through transcribed RNA or translated protein). In general, each mRNA molecule goes on to make a specific protein (or set of proteins). In some cases this protein will be structural, and will accumulate at the cell membrane or within the cell to give it particular structural properties. In other cases the

© 2019 IJSRET

protein will be an enzyme, i.e., a micro-machine that catalyses a certain reaction, such as the breakdown of a food source or toxin. Some proteins though serve only to activate other genes, and these are the transcription factors that are the main players in regulatory networks or cascades. By binding to the promoter region at the start of other genes they turn them on, initiating the production of another protein, and so on. Some transcription factors are inhibitory.

In single-celled organisms, regulatory networks respond to the external environment, optimizing the cell at a given time for survival in this environment. Thus a yeast cell, finding itself in a sugar solution, will turn on genes to make enzymes that process the sugar to alcohol. This process, which we associate with wine-making, is how the yeast cell makes its living, gaining energy to multiply, which under normal circumstances would enhance its survival prospects.

In multicellular animals the same principle has been put in the service of gene cascades that control body-shape. Each time a cell divides, two cells result which, although they contain the same genome in full, can differ in which genes are turned on and making proteins. Sometimes a 'self-sustaining feedback loop' ensures that a cell maintains its identity and passes it on. Less understood is the mechanism of epigenetic by which chromatin modification may provide cellular memory by blocking or allowing transcription. A major feature of multi cellular animals is the use of morphogen gradients, which in effect provide a positioning system that tells a cell where in the body it is, and hence what sort of cell to become.

A gene that is turned on in one cell may make a product that leaves the cell and diffuses through adjacent cells, entering them and turning on genes only when it is present above a certain threshold level. These cells are thus induced into a new fate, and may even generate other morphogens that signal back to the original cell.

Over longer distances morphogens may use the active process of signal transduction. Such signaling controls embryogenesis, the building of a body plan from scratch through a series of sequential steps. They also control and maintain adult bodies through feedback processes, and the loss of such feedback because of a mutation can be responsible for the cell proliferation that is seen in cancer.

In parallel with this process of building structure, the gene cascade turns on genes that make structural proteins that give each cell the physical properties it needs. The nodes of this network can represent genes, proteins, mRNAs, protein/protein complexes or cellular processes. Nodes that are depicted as lying along vertical lines are associated with the cell/environment interfaces, while the others are free-floating and can diffuse.

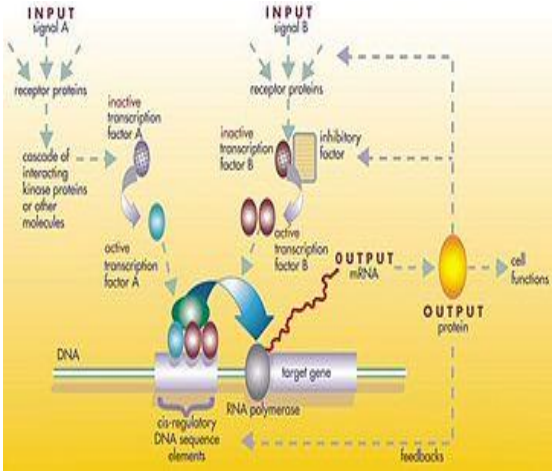


Fig. 1 Structure of a gene regulatory network.

A typical gene regulatory network looks something like this:

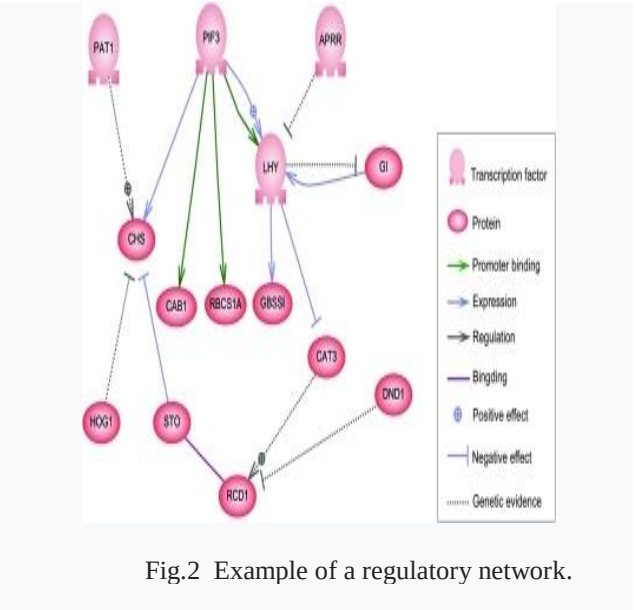


Fig.2 Example of a regulatory network.

Edges between nodes represent interactions between the nodes that can correspond to individual molecular reactions between DNA, mRNA, miRNA, proteins or molecular processes through which the products of one gene affect those of another, though the lack of experimentally obtained information often implies that some reactions are not modeled at such a fine level of detail.These interactions can be inductive (usually represented by arrowheads or the + sign), with an increase in the concentration of one leading to an increase in the other, inhibitory (represented with filled circles, blunt

© 2019 IJSRET

arrows or the minus sign), with an increase in one leading to a decrease in the other, or dual, when depending of the circumstances the regulator can activate or inhibit the target node. The nodes can regulate themselves directly or indirectly, creating feedback loops, which form cyclic chains of dependencies in the topological network. The network structure is an abstraction of the system's molecular or chemical dynamics, describing the manifold ways in which one substance affects all the others to which it is connected. In practice, such GRNs are inferred from the biological literature on a given system and represent a distillation of the collective knowledge about a set of related biochemical reactions. To speed up the manual curation of GRNs, some recent efforts try to use text mining, curated databases, network inference from massive data, model checking and other information extraction technologies for this purpose.

II. RELATED WORK

Using gene expression data to refer gene regulatory networks (GRNs) is a key approach to understand the relationships between transcription factors (TFs) and target genes that may aid to uncover underneath biochemical pathways governed by the TFs. Analyzing individual TF and gene associations to the level that induces biological significance through wet-lab experiments is a practically challenging, costly and time-consuming task (Pe'er & Hacohen, 2011). It is therefore useful to adopt computational methods to obtain similar information, because results obtained from such methods can easily and quickly be reproduced through inexpensive procedures while allowing multiple methods to explore data to validate outcomes (Cerulo et al., 2010; Gillani et al., 2014; Lingeman & Shasha, 2012; Maetschke et al., 2014).

Various computational methods for performing unsupervised, supervised and semisupervised prediction of GRNs have been proposed. These methods employ a variety of techniques ranging from Boolean networks (Lähdesmäki et al., 2003) and Bayesian networks (Acerbi et al., 2014; Vignes et al., 2011) to compressive sensing (Chang et al., 2014). Integrated toolkits combining different network inference methods are also available (Hurley et al., 2015). Many of the methods are unsupervised. In previous studies, several authors have shown that supervised and semi-supervised methods outperformed unsupervised methods (Cerulo et al., 2010; Maetschke et al., 2014).

However, supervised methods require training data to contain both positive and negative examples, which are difficult to obtain. In contrast, semi-supervised methods can work with a large number of unlabeled examples, which are much easier to obtain (Elkan & Noto, 2008). In this paper, a new semi-supervised method has been proposed that is capable of predicting TF-gene interactions in the presence of unlabeled training examples. In order to predict GRNs it is essential to understand possible TF-gene interaction types. Figure 3 illustrates a gene regulatory network between the Escherichia coli transcription factor FNR and several target genes. The network is created with Cytoscape (Shannon et al., 2003) using true TF-gene interactions obtained from RegulonDB (Salgado et al., 2013). In general, if there exists an experimentally verified interaction between a TF and a target gene, then such an interaction is considered to be known. The known interactions are generated through wet-lab or sometimes dry-lab experiments that are indirectly associated with wet-lab experiments, and curated based on experimental outputs.

On the contrary, TF-gene interactions that are not yet experimentally verified are considered to be unknown. In Figure 3, solid edges represent known interactions and dotted edges represent unknown interactions. There are two types of known interactions: up-regulation and down-regulation. If a transcription factor activates a gene, then the gene is up-regulated. If a transcription factor inhibits (or represses) a gene, then the gene is down-regulated.

1. Network prediction

There are three types of computational methods for predicting gene regulatory networks (GRNs); they are supervised, semi-supervised, and unsupervised methods (Maetschke et al., 2014). The first two types of methods differ primarily based on whether training examples are labeled. In supervised methods, each training example must have a (positive or negative) label In semi-supervised methods, some training examples are labeled and some are unlabeled. The training examples for the supervised and semi-supervised methods are obtained from known and unknown TF-gene interactions as described earlier. On the other hand, there is no concept of training for unsupervised methods.

Specifically, in the supervised methods, the training set contains both positive and negative examples. Known interactions are used as positive examples. However, obtaining negative examples is a challenging task, due to the lack of biological evidence to claim that there is no regulatory connection between a transcription factor and a target gene (Gillani et al., 2014). Some authors (Gillani et al., 2014; Mordelet & Vert, 2008) assumed unknown interactions to be negative examples.

However, as explained in Section 1.1, these unknown interactions are not yet verified experimentally; some of them may turn out to be positive examples. For instance, in the first release of RegulonDB (1.0), 533 regulatory interactions were identified. This number was increased to 4,268 in the later release of RegulonDB (8.0), meaning

© 2019 IJSRET

that at the time of the first release 3,735 interactions were unknown, which later turned out to be actually valid interactions (Huerta et al., 1998; Salgado et al., 2013). Under this circumstance, a semi-supervised method is more suitable, which treats all unknown interactions as unlabeled examples and used positive and unlabeled examples to predict gene regulatory networks (GRNs).

The purpose of this study is to investigate semi-supervised methods for GRN prediction. Four transcription factors have been considered from E. coli, namely ARCA, CRP, FIS and FNR. Similarly four transcription factors from S. cerevisiae have been chosen namely REB1, ABF1, CBF1 and GCN4. The four specific transcription factors (TFs) were chosen because they had the largest numbers, ranging from 100 to 400, of known interactions with target genes in the respective organisms. These known interactions were used as positive examples in this study. By using an iterative procedure, unlabeled examples has been refined (i.e., unknown interactions) at hand to get more reliable negative examples for all the four TFs for both organisms.

Inductive and transductive learning approaches has been adopted for GRN prediction using the semi-supervised methods. With the inductive learning approach, a model is learned from a training dataset, and the model is then applied to a separate testing dataset that is disjoint from the training dataset; hence any information concerning the test data is not seen or used while creating the model (Mitchell, 1997). On the contrary, the transductive learning approach builds a model based on both the training data and some information from the testing dataset, and the model is then applied to the test data (Vapnik, 1998).Two machine learning algorithms have been employed, namely support vector machines (SVM) and random forests (RF), in this study. The prediction accuracies of both algorithms for the chosen transcription factors of E. coli and S. cerevisiae were calculated and compared.

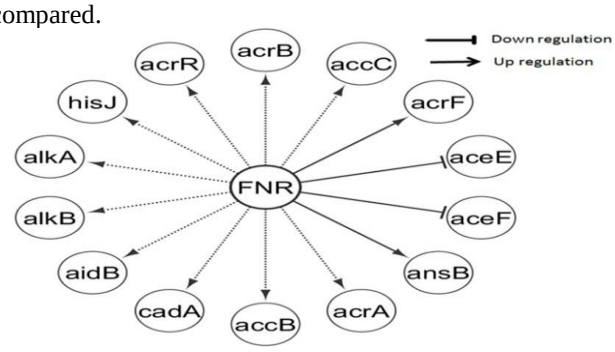


Fig. 3 Diagram showing the true regulatory relationships between the E.

coli transcription factor FNR and several target genes. As annotation implies, arrows with solid lines represent up regulations and inverted T shape solid lines represent down-regulations. These up- and down regulations together comprise positive examples in this study. Arrows with dotted lines represent unknown interactions, which serve as unlabeled examples in this study.

2. Datasets

Marbach et al. (2012) performed a comprehensive assessment of network inference methods through the DREAM5 project on both E. coli and S. cerevisiae. Their study was conducted using two E. coli regulatory databases: EcoCyc accessible at http://ecocyc.org/ (Keseler et al., 2013) and RegulonDB accessible at http://regulondb.ccg.unam.mx/ (Salgado et al., 2013). RegulonDB was used in this study because it is a popular database for benchmark experiments.

The latest version (version 8.6) of RegulonDB contains 4,268 known TF-gene interactions; these interactions were obtained from the E. coli K 12 strand (Salgado et al., 2013). Hence we used the gene expression datasets specifically generated from E. coli K 12. These gene expression datasets had GEO accession numbers GSE21869 (Asakura et al., 2011), GSE10158 (Laubacher & Ades, 2008), GSE12411 (Aggarwal & Lee, 2011), GSE33147 (Fong et al., 2005), and GSE17505 (Haddadin & Harcum, 2005). All the datasets are freely available at Gene Expression Omnibus (GEO) (http://www.ncbi.nlm.nih.gov/geo/) and were produced with Affymetrix E. coli Antisense Genome Array that contains 7,312 probe sets for E. coli gene expression analysis. For S. cerevisiae, there were three regulatory databases used by the DREAM5 study (Marbach et al., 2009, 2010, 2012; Prill et al., 2010), which included YEASTRACT accessible at http://www.yeastract.com/ (Abdulrehman et al., 2011),

Fraenkel et al.'s map of conserved regulatory sites accessible at http://fraenkel.mit.edu/improved_map/ (MacIsaac et al., 2006), and the database described in (Hu et al., 2007). The DREAM5 study evaluated these databases and reported that Fraenkel's database contains high quality TF-gene interactions; consequently these interactions has been used as positive examples for S. cerevisiae in this study. Five gene expression datasets for S. cerevisiae have been chosen. These datasets had GEO accession numbers GSE30052 (Chin et al., 2012), GSE12221 (Shalem et al., 2008), GSE12222 (Shalem et al., 2008), GSE40817 (Yona et al., 2012), and GSE8799 (Orlando et al., 2008). All the yeast datasets were created using Affymetrix Yeast Genome 2.0 Array containing 5,744 probe sets for S. cerevisiae gene expression analysis. Expression vectors of TFs has been extracted and genes that were present in Regulon DB and Fraenkel's database respectively, and created an

© 2019 IJSRET

expression matrix containing the expression vectors for each of the gene expression datasets mentioned above. The E. coli expression matrices contained 1,161 gene expressions vectors and the S. cerevisiae expression matrices contained 1,994 gene expressions vectors. These matrices were then scaled to zero mean and unit standard deviation. As explained in earlier section, positive examples were created using the known interactions found in RegulonDB and Fraenkel's database for E. coli and S. cerevisiae respectively. To obtain unknown interactions, all possible combinations of available TF and gene pairs have been generated.

Each of these combinations was considered as an unknown interaction provided that it did not exist in RegulonDB (Fraenkel's database, respectively) for E. coli (S. cerevisiae, respectively). These unknown interactions were treated as unlabeled examples. Then all the interactions were separated based on the TFs. For each organism, the top four TFs that had the largest number of interactions were chosen and used to perform the experiments. Table 1 lists the number of positive and unlabeled examples for each chosen TF of E. coli and S. cerevisiae respectively. The columns named Positive in the table show the total number of known interactions for each TF in E. coli and S. cerevisiae respectively.

Table 1. The number of positive and unlabeled examples for each transcription factor of E. coli and S. cerevisiae respectively used in this study

E. coli			S. cerevisiae		
TF	Positive	Unlabeled	TF	Positive	Unlabeled
CRP	390	770	REB1	217	1776
FNR	239	921	FEB1	199	1794
FIS	200	960	CBF1	164	1829
ARCA	139	1021	GCN4	120	1873

Both supervised and semi-supervised methods work under the principle that if a gene is known to interact with another gene, then any other two genes containing similar gene expression profiles are also likely to interact with each other (Cerulo et al., 2010; Mordelet & Vert, 2008). Based on this principle, feature vectors for TFs and genes were constructed by concatenation of their expression profiles.

Hence the resulting feature vectors contained twice the number of features than the original individual expression vectors. While concatenating two profiles, orders were considered, which means, if G1 is known to regulate G2 then the feature vector V (G1, G2) can only be created such that expression values of G1 are followed by expression values of G2. In other words, the feature vector V (G1, G2) implies that G1 regulates G2 but the opposite of that may not necessarily be true. After concatenation, the resulting feature vectors were scaled to zero mean and unit standard deviation.

III. PROPOSED SEMI-SUPERVISED METHODS

Two machine learning algorithms, namely support vector machines (SVM) and random forests (RF), were employed for making predictions. SVM analysis was done using the LIBSVM package in R, accessible at http://cran.r-project.org/web/packages/e1071/index.html (Chang & Lin, 2011). RF analysis was performed using the random Forest package in R, accessible at http://cran.r project.org/web/packages /random Forest /index.html (Liaw & Wiener, 2002). Figure 3 presents the flowchart of the proposed semi-supervised methods for predicting gene regulatory networks (GRNs). Initially there is positive and unlabeled examples, where positive examples represent known TF-gene interactions and unlabeled examples represent unknown interactions. These examples are converted to feature vectors using the method described in Section 2.1, where the feature vectors are used as input of the SVM and RF algorithms. An iterative procedure is then executed; in each iteration a classification model is obtained and validated.

In Figure 3, Positive is the set of all positive examples obtained from known interactions of each individual transcription factor of an organism, Unlabeled is the set of all unlabeled examples for the organism, P represents the positive training dataset, N represents the negative training dataset, V represents the validation dataset, and T represents the testing dataset. The validation set V contains only positive examples, since true negative examples are not available. All the predictions are made on the examples in V, and prediction accuracies of these methods are calculated by comparing true labels with predicted labels. The Positive set is evenly divided into two disjoint subsets, P and V, where P and V contain approximately the same number of positive.

Examples. For a given transcription factor (TF), P and V remain the same throughout all the iterations for both inductive and transductive learning approaches. T comprises only unlabeled examples, which are used in the iterative procedure to produce reliable negative examples. Refer to Figure 3. During iteration k, $0 \leq k \leq K$, the SVM or RF algorithm is trained using P and Nk. A binary classification model, denoted Model k , is obtained. This (SVM or RF) model is then applied to the validation dataset V to predict the labels of the examples in V. The prediction accuracy of the model is calculated by dividing the number of correctly predicted examples in V by the

© 2019 IJSRET

total number of examples in V. In addition, this model, Modelk, is also applied to the testing set Tk to classify the unlabeled examples in Tk as shown in Figure 2. Both SVM and RF algorithms are able to assign probabilistic weights to their classification results. sing these probability values we extract reliable negative examples from the set T by choosing the bottom |N| unlabeled examples in T that have the lowest probability of being positive.

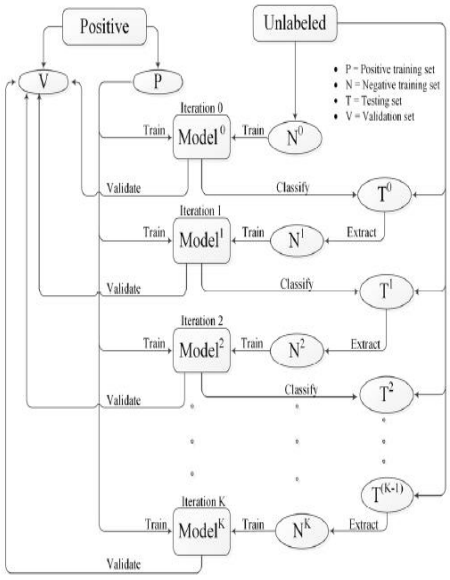


Fig. 4 Flowchart of the proposed semi-supervised GRN prediction methods

These |N| negative examples are collected and stored in Nk+1, which, together with P, will be used to train the SVM or RF algorithm in iteration k+1.It has been noticed that iteration 0 is a special case, in which randomly |N| examples have been selected from the Unlabeled set and store them in N0. This has been also noticed that this is a balanced binary classification, since |P| = |N| throughout all the iterations. The difference between the inductive and transductive learning approaches is on whether the sets N and T are disjoint.

For the inductive learning approach, T is created such that N and T are disjoint. For the transductive learning approach, however, T includes all the available unlabeled examples (i.e., all the examples in the Unlabeled set). More precisely, in iteration k, for the inductive learning approach, Tk = Unlabeled – Nk and Nk ∩ Tk = Ø; for the transductive learning approach, Tk = Unlabeled and Nk is the subset of Tk. The iterative procedure is executed for each of the four chosen transcription factors (TFs) in each organism. Table 2 lists the number of examples in P, N, T and V for each TF in E. coli and Table 3 shows the data for each TF in S. cerevisiae. The Total column in each table shows the total number of training examples, which is equal to |P| + |N|, used in each iteration. The Inductive (Transductive, respectively) column in each of these tables shows the number of unlabeled examples in the testing set T used by the inductive (transductive, respectively) learning approach.

Table 2 Number of examples in P, N, T and V respectively for each TF in E. coli

TF	P	N	Total	T		V
				Inductive	Transductiuve	
CRP	200	200	400	580	775	200
FNR	125	125	250	806	926	125
FIS	105	105	210	865	965	105
ARCA	75	75	150	956	1026	74

Table 3 Number of examples in P, N, T and V respectively for each TF in S.cerevisiae

TF	P	N	Total	T		V
				Inductive	Transductiuve	
REB1	113	113	226	1673	1781	114
FEB1	105	104	209	1700	1799	105
CBF1	87	87	174	1752	1834	87
GCN4	65	65	130	1818	1875	65

In all the experiments, the radial basis function (RBF) kernel was used for support vector machines (SVM) with all other parameters set to default values. With random forests (RF), all parameters were set to default values and the number of trees used was 505. For statistical consistency and fair comparisons, the Positive and Unlabeled sets were kept exactly the same for both SVM and RF in the initial iteration (i.e., iteration 0) for each choosen TF in E. coli and S.cerevisiae.

IV. RESULTS AND DISCUSSION

A series of experiments has been carried out to evaluate the performance of the proposed semi supervised methods on the different datasets described in Section 2.2, where the performance of a method was measured by the prediction accuracy of the method. Here, the prediction accuracy of a method is defined as the number of correctly predicted examples in a validation dataset divided by the total number of examples in the validation dataset (cf. Figure 3). Figure 4 compares the transductive and inductive learning approaches with the SVM and RF, algorithms respectively, where the experimental results were obtained using the E. coli transcription factor ARCA and dataset GSE21869. Figure 6 compares the transductive and inductive learning approaches with the

© 2019 IJSRET

SVM an RF algorithm, where the experimental results were obtained using the S. cerevisiae transcription factor REB1 and dataset GSE12222. It can be seen from Figure 5 and Figure 6 shows that, for SVM, the transductive learning approach yielded higher prediction accuracies than the inductive learning approach. X-Axis → Number of iterations, Y-Axis→Accuracy

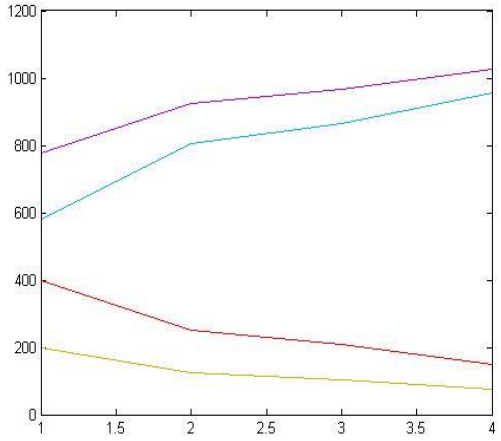


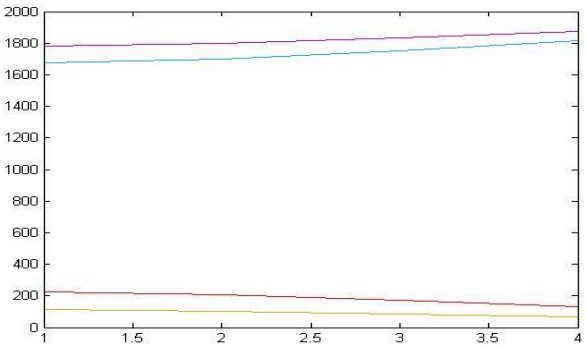
Fig. (5) Performance comparison of the transductive and inductive learning approaches based on the E. coli transcription factor ARCA and dataset GSE21869 with (a) the SVM algorithm, and(b) the RF algorithm.

The performance of the transductive learning approach became stable as the number of iterations in the proposed semi-supervised methods increased. On the other hand, the performance of the inductive learning approach tended to fluctuate up and down with high frequency. RF exhibited a similar pattern as far as the relative performance of the transductive learning and inductive learning approaches was concerned. It is worth pointing out that the accuracies of the proposed semi-supervised methods were relatively low when only one iteration (i.e., iteration 0) was executed. Refer to Figure 4. In iteration 0, the negative training set N0 was comprised of randomly selected unlabeled examples.

On the other hand, starting from the second iteration (i.e., iteration 1), the proposed semi supervised methods picked unlabeled examples that had the lowest probability of being positive and used them as negative training examples. These unlabeled examples with the lowest probability of being positive formed more reliable negative training data than the randomly selected unlabeled examples, hence yielding higher accuracies. In subsequent experiments, the transductive learning approach has been adopted and fixed the number of iterations at 20. Semi-supervised GRN prediction methods have been applied, with both SVM and RF, to the different gene expression datasets for the different transcription factors selected from E. coli and S. cerevisiae respectively. Figure (7) shows a bar chart comparing the performance of SVM and RF using the gene expression datasets GSE10158, GSE12411, GSE33147, GSE21869 and GSE17505 for the E. coli transcription factor ARCA (CRP, FIS, FNR respectively). Figure (8) shows a bar chart comparing the performance of SVM and RF using the gene expression datasets GSE30052, GSE12221, GSE12222, GSE40817 and GSE8799 for the S. cerevisiae transcription factor REB1 (ABF1, CBF1, GCN4 respectively). It can be seen from Figure 5 that SVM yielded higher prediction accuracies than RF on the datasets GSE33147 and GSE17505 for the E. coli transcription factor ARCA while RF performed better than SVM on the dataset GSE10158.

The two machine learning algorithms exhibited similar performance on the datasets GSE12411 and GSE21869. Figure 5 shows that SVM and RF exhibited similar performance on all datasets except GSE17505, where SVM was more accurate than RF for the transcription factor CRP. Significant discrepancies were observed in the predictions accuracies with the transcription factor FIS, where RF outperformed SVM on all datasets except GSE33147 (Figure5).

Finally, as Figure 5 implies, SVM and RF did not show any major difference in GRN prediction for the E. coli transcription factor FNR. Figure 6 shows that the performance of SVM and RF was nearly identical across all the datasets for all the S. cerevisiae transcription factors used in this study. There were only two instances where significant differences were observed. The first instance was on the dataset GSE8799 for the S. cerevisiae transcription factor ABF1 (Figure 6). The second instance was on the dataset GSE12222 for the transcription factor GCN4 (Figure 6). In both instances, RF performed better than SVM.



X-Axis → Number of iterations, Y-Axis→Accuracy

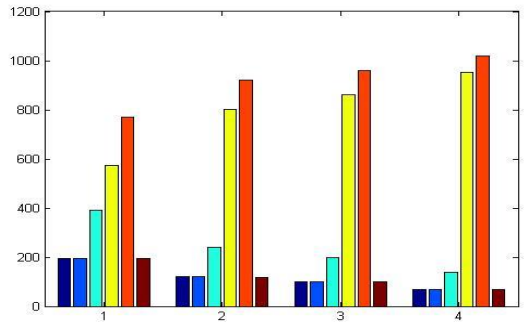
Fig. (6) Performance comparison of the transductive and inductive learning approaches based on the S. cerevisiae transcription factor REB1 and dataset GSE12222 with (a) the SVM algorithm, and (b) the RF algorithm.

© 2019 IJSRET

The performance of the transductive learning approach became stable as the number of iterations in the proposed semi-supervised methods increased. On the other hand, the performance of the inductive learning approach tended to fluctuate up and down with high frequency. RF exhibited a similar pattern as far as the relative performance of the transductive learning and inductive learning approaches was concerned. It is worth pointing out that the accuracies of the proposed semi-supervised methods were relatively low when only one iteration (i.e., iteration 0) was executed. Refer to Figure4. In iteration 0, the negative training set N0 was comprised of randomly selected unlabeled examples.

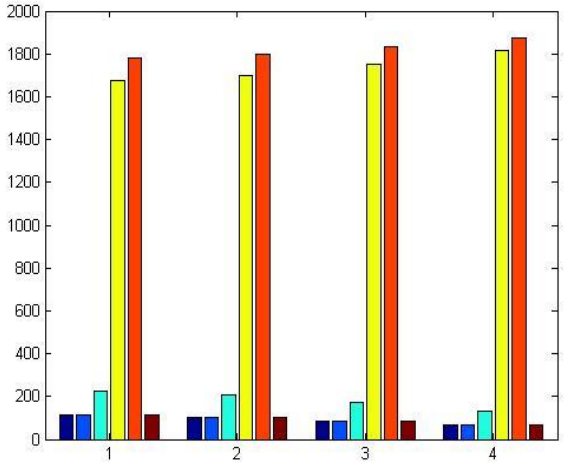
On the other hand, starting from the second iteration (i.e., iteration 1), the proposed semi supervised methods picked unlabeled examples that had the lowest probability of being positive and used them as negative training examples. These unlabeled examples with the lowest probability of being positive formed more reliable negative training data than the randomly selected unlabeled examples, hence yielding higher accuracies. In subsequent experiments, we adopted the transductive learning approach and the number of iterations at 20.

Semi-supervised GRN prediction methods has been applied, with both SVM and RF, to the different gene expression datasets for the different transcription factors selected from E. coli and S. cerevisiae respectively. Figure 7 shows a bar chart comparing the performance of SVM and RF using the gene expression datasets GSE10158, GSE12411, GSE33147, GSE21869 and GSE17505 for the E. coli transcription factor ARCA (CRP, FIS, FNR respectively). Figure 8 shows a bar chart comparing the performance of SVM and RF using the gene expression datasets GSE30052, GSE12221, GSE12222, GSE40817 and GSE8799 for the S. cerevisiae transcription factor REB1 (ABF1, CBF1, GCN4 respectively).



X-Axis-> Data set, Y-Axis -> Accuracy

Fig.(7) Performance comparison of the SVM and RF algorithms with the transductive learning approach on five gene expression datasets GSE10158, GSE12411, GSE33147, GSE21869 and GSE17505, and four transcription factors of E. coli including (a) ARCA, (b) CRP, (c) FIS, and (d) FNR. Dark bars represent SVM and light bars represent RF.



X-Axis-> Data set, Y-Axis -> Accuracy

Fig 8**.** Performance comparison of the SVM and RF algorithms with the transductive learning approach on five gene expression datasets GSE30052, GSE12221, GSE12222, GSE40817 and GSE8799, and four transcription factors of S. cerevisiae including (a) REB1, (b) ABF1, (c) CBF1, and (d) GCN4. Dark bars represent SVM and light bars represent RF.

It can be seen from Figure 7 that SVM yielded higher prediction accuracies than RF on the datasets GSE33147 and GSE17505 for the E. coli transcription factor ARCA while RF performed better than SVM on the dataset GSE10158. The two machine learning algorithms exhibited similar performance on the datasets GSE12411 and GSE21869. Figure 7 shows that SVM and RF exhibited similar performance on all datasets except GSE17505, where SVM was more accurate than RF for the transcription factor CRP.

Significant discrepancies were observed in the predictions accuracies with the transcription factor FIS, where RF outperformed SVM on**)** All datasets except GSE33147 (Figure5). Finally, as Figure 7 implies, SVM and RF did not show any major difference in GRN prediction for the E. coli transcription factor FNR. Figure 8 shows that the performance of SVM and RF was nearly identical across all the datasets for all the S. cerevisiae transcription factors used in this study. There were only two instances where significant differences were observed. The first instance was on the dataset GSE8799 for the S. cerevisiae transcription factor ABF1 (Figure 8). The second instance

© 2019 IJSRET

was on the dataset GSE12222 for the transcription factor GCN4 (Figure 8). In both instances, RF performed better than SVM. For each of the organisms E. coli and S. cerevisiae, four transcription factors have been chosen having enough known TF-gene interactions, and evaluated the effectiveness of the proposed semi-supervised methods. These methods employed an iterative procedure together with a transductive learning strategy or an inductive learning strategy to obtain more reliable negative training examples.

Experimental results indicated that the transductive learning approach consistently outperformed the inductive learning approach on the datasets tested in this study. The results also showed that after certain iterations, the prediction accuracy of the transductive learning approach tended to converge. For many experiments the convergent point was identified within the first 20 interactions. To provide more reliable and consistent findings the number of iterations have been fixed to at 20 in performing the rest of the experiments. With 20 iterations, no significant difference in prediction accuracies was observed between the SVM and RF algorithms using the transductive learning approach. On the other hand, the inductive leaning approach did not exhibit a clear convergent point; instead, it exhibited waving patterns; cf.

Figure 4. Such a behavior might exist due to the fact that in the inductive learning approach a portion of TF-gene interactions were not used. As explained in Section 2.2, in iteration k, Tk = Unlabeled – Nk, and we extract some unlabeled examples from Tk to get negative training examples in Nk+1 that are used in iteration k+1. Tk does not contain the TF-gene interactions in Nk. Hence when the TF-gene interactions in Tk are ranked in the descending order of their probabilities of being positive, they do not contain the TF-gene interactions in Nk, and therefore these TF-gene interactions in Nk will not be in Nk+1. It is likely that Nk may contain some TF-gene interactions that are very reliable negative examples.

As a consequence, these very reliable negative examples will not be in Nk+1 used for training the machine learning algorithms (SVM or RF) in iteration k+1. Hence when those very reliable negative examples are in the training set, the trained model yields a high accuracy; otherwise the trained model yields a low accuracy. This explains why the inductive learning approach exhibited waving patterns in terms of prediction accuracies. Note that, in the transductive learning approach, Tk = Unlabeled, and hence those very reliable negative examples are always considered in every iteration. As a consequence, the performance of the transductive learning approach became stable when sufficient high-quality negative examples were collected after a certain number of iterations (e.g., after 20 iterations). The experimental results in Section 3 were obtained using the radial basis function (RBF) kernel for the SVM algorithm and 500 trees for the RF algorithm. The linear kernel and polynomial kernel available in the LIBSVM package has been also tasted with default parameter values on all five gene expression datasets and for all four transcription factors. RF algorithm, has been tested using different numbers of trees, specifically 100, 500 and 1,000 trees respectively. The number of iterations used in the semi-supervised methods was fixed at 20. The results obtained were similar to those presented in Section 3.

These results indicate that all the three kernels, namely RBF, linear and polynomial kernels, work well. Furthermore, the number of decision trees used in the RF algorithm has little impact on its performance provided the number is sufficiently large (e.g., at least 100). However, with too many trees (e.g., the number of trees is greater than 1,000), the time of the RF algorithm may increase substantially since it takes a large amount of time to build these trees. Refer to Figures 4 and 5. The performance of the SVM algorithm clearly converged in both organisms with the transductive learning strategy in the sense that after a certain number of iterations (e.g., after 20 iterations), the accuracies of the algorithm did not vary too much.

The reason behind this phenomenon is that, after 20 iterations, the algorithm has identified the most reliable negative training examples, which remain the same for subsequent iterations. Therefore the models created for those subsequent iterations by the reliable negative training examples and the positive training set P remain almost the same, and hence always make the same predictions; cf. Figure 5 and Figure 6). On the other hand, referring to Figure 5 and Figure 6, we see that, with the transductive learning strategy, there were slight variations in the performance of the RF algorithm even after 20 iterations were executed. Although those variations are negligible, allowing us to draw the qualitative conclusions based on our findings, a close look at the machine learning algorithms explains why the variations occur.

In general, the SVM algorithm systematically attempts to find a hyperplane that maximizes the distance to the nearest training example of any class (Joachims, 1999). There is no randomness associated with the SVM algorithm. On the contrary, the RF algorithm randomly picks training examples and features in the training examples to build decision trees (Breiman, 2001). Due to the randomness associated with the RF algorithm, the strong convergence was not observed for the algorithm; cf. Figure 5 and Figure 6 It is worth pointing out that the proposed semi-supervised methods performed better than the supervised methods described in (Gillani et al., 2014; Mordelet & Vert, 2008). Just like how the semi-

© 2019 IJSRET

supervised methods work in the first iteration (i.e., iteration 0) where randomly chosen unlabeled examples in N0 and positive examples in P are used to train a machine learning algorithm (e.g., SVM), the supervised methods treat the unlabeled examples as negative examples and use them together with the positive examples in P to train the SVM algorithm (Gillani et al., 2014; Mordelet & Vert, 2008). As shown in Figures 5 and 6, executing merely the first iteration without iteratively refining the unlabeled examples to obtain more reliable negative training examples performs worse than executing several (e.g., 20) iterations as done by the proposed semi-supervised methods, suggesting that the proposed semi-supervised methods be better than the supervised methods.

V. CONCLUSIONS

The idea of training a classifier using positive and unlabeled examples was previously proposed to classify web pages (Blum & Mitchell, 1998) and text documents (Li & Liu, 2003; Liu et al., 2003).This idea has been extended here to predict gene regulatory networks using both inductive and transductive learning approaches. To utilize available unlabeled examples and to effectively extract reliable negative examples, support vector machines (SVM) and random forests (RF), both of which were able to assign probabilistic weights to their classification results. Negative examples from the testing set has been picked that had the lowest probability of being positive.

Experimental results showed that the negative examples chosen this way yielded better performance than the negative examples that were randomly selected from the unlabeled data. In addition, this results showed that the transductive learning approach outperformed the inductive learning approach and exhibited a relatively stable behavior for both SVM and RF algorithms on the datasets used in this study. Algorithmic parameters such as different kernels for SVM and different numbers of decision trees for RF did not yield significantly different results. Furthermore, there was no clear difference in the performance of SVM and RF for both the prokaryotic organism (E. coli) and the eukaryotic organism (S. cerevisiae).

The experimental results also showed that our proposed semi-supervised methods were more accurate than the supervised methods described in (Gillani et al., 2014; Mordelet & Vert, 2008) since the semi-supervised methods adopted an iterative procedure to get more reliable negative training examples than those used by the supervised methods. A shortcoming of semi-supervised methods is that, sometimes not enough known TF-gene interactions are available for certain transcription factors, or there exists no known TFgene interaction at all for some organisms. Under this circumstance, semi-supervised methods like what we describe here may yield low prediction accuracies due to the lack of reliable training data. One possible way for dealing with organisms with only unknown TF-gene interactions or unlabeled examples is to use SVM and RF algorithms to assign probabilistic weights to their classification results. Then pick positive (negative, respectively) examples from the testing set that have the highest (lowest, respectively) probability of being positive, and use these more reliable positive and negative training data to obtain a hopefully better classification model. In future work, we plan to investigate the performance of this approach and compare different machine learning algorithms using this approach. Another line of future work is to compare semi-supervised and supervised methods using simulated data.Further research will be conducted to develop additional biologically meaningful features and to evaluate the effectiveness of those features.

REFERENCES

1. Abdulrehman D, Monteiro PT, Teixeira MC, Mira NP, Lourenço AB, dos Santos SC, Cabrito TR, Francisco AP, Madeira SC, Aires RS, Oliveira AL, Sá-Correia I, Freitas AT. YEASTRACT: providing a programmatic access to curated transcriptional regulatory associations in Saccharomyces cerevisiae through a web services interface. Nucleic Acids Res. 2011;39(Database issue):D136-40.
2. Acerbi E, Zelante T, Narang V, Stella F. Gene network inference using continuous time Bayesian networks: a comparative study and application to Th17 cell differentiation. BMC Bioinformatics 2014;15:387.
3. Aggarwal K, Lee KH. Overexpression of cloned RhsA sequences perturbs the cellular translational machinery in Escherichia coli. J. Bacteriol. 2011;193(18):4869-80.
4. Asakura Y, Kojima H, Kobayashi I. Evolutionary genome engineering using a restriction modification system. Nucleic Acids Res. 2011;39(20):9034-46. Florian Leitner, Martin Krallinger, Sushil Tripathi, Martin Kuiper, Astrid Lægreid and Alfonso Valencia, Mining cis-Regulatory Transcription Networks from Literature, Proceedings of BioLINK Special Interest Group, 5–12, ISBM/ECCB, 2013
5. Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. In Proceedings of the 11th Annual ACM Conference on Computational Learning Theory, 1998. Breiman L. Random forests. Machine Learning 2001;45(1):5-32.
6. Cerulo L, Elkan C, Ceccarelli M. Learning gene regulatory networks from only positive and unlabeled data. BMC Bioinformatics 2010;11:228.

© 2019 IJSRET

7. Chang C, Lin C. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2011;2(3):27.
8. Chang YH, Gray JW, Tomlin CJ. Exact reconstruction of gene regulatory networks using compressive sensing. BMC Bioinformatics 2014;15:400.
9. Chin SL, Marcus IM, Klevecz RR, Li CM. Dynamics of oscillatory phenotypes in Saccharomyces cerevisiae reveal a network of genome-wide transcriptional oscillators. FEBS J. 2012;279(6):1119-30.
10. Elkan C, Noto K. Learning classifiers from only positive and unlabeled data. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008. Fong SS, Joyce AR, Palsson BØ. Parallel adaptive evolution cultures of Escherichia coli lead to convergent growth phenotypes with different gene expression states. Genome Res. 2005;15(10):1365-72.
11. Gillani Z, Akash MS, Rahaman MD, Chen M. CompareSVM: supervised, Support Vector Machine (SVM) inference of gene regularity networks. BMC Bioinformatics 2014;15:395. Haddadin FT, Harcum SW. Transcriptome profiles for high-cell-density recombinant and wildtype Escherichia coli. Biotechnol. Bioeng. 2005;90(2):127-53.
12. Hu Z, Killion PJ, Iyer VR. Genetic reconstruction of a functional transcriptional regulatory network. Nature Genetics 2007;39(5):683-7.
13. Huerta AM, Salgado H, Thieffry D, Collado-Vides J. RegulonDB: a database on transcriptional regulation in Escherichia coli. Nucleic Acids Res. 1998;26(1):55-9.
14. Hurley DG, Cursons J, Wang YK, Budden DM, Print CG, Crampin EJ. NAIL, a software toolset for inferring, analyzing and visualizing regulatory networks. Bioinformatics 2015;31(2):277-8.
15. Joachims T. Making large-scale support vector machine learning practical. Advances in Kernel Methods, MIT Press, pages 169-84, 1999.
16. Keseler IM, Mackie A, Peralta-Gil M, Santos-Zavaleta A, Gama-Castro S, Bonavides-Martínez C, Fulcher C, Huerta AM, Kothari A, Krummenacker M, Latendresse M, Muñiz-Rascado L, Ong Q, Paley S, Schröder I, Shearer AG, Subhraveti P, Travers M, Weerasinghe D, Weiss V, Collado-Vides J, Gunsalus RP, Paulsen I, Karp PD. EcoCyc: fusing model organism databases with systems biology. Nucleic Acids Res. 2013;41(Database issue):D605-12.
17. Lähdesmäki H, Shmulevich I, Yli-Harja O. On learning gene regulatory networks under the Boolean network model. Machine Learning 2003;52(1-2):147-67.
18. Laubacher ME, Ades SE. The Rcs phosphorelay is a cell envelope stress response activated by peptidoglycan stress and contributes to intrinsic antibiotic resistance. J. Bacteriol. 2008;190(6):2065-74.
19. Li XL, Liu B. Learning to classify texts using positive and unlabeled data. In Proceedings of the 18th International Joint Conference on Artificial Intelligence, pages 587-92, 2003. Liaw A, Wiener M. Classification and regression by randomForest. R News 2002;2(3):18-22.
20 .Lingeman JM, Shasha D. Network Inference in Molecular Biology: A Hands-on Framework. Springer, 2012.
21. Liu B, Dai Y, Li X, Lee WS, Yu PS. Building text classifiers using positive and unlabeled examples. In Proceedings of the 3rd IEEE International Conference on Data Mining, 2003.
22. MacIsaac KD, Wang T, Gordon DB, Gifford DK, Stormo GD, Fraenkel E. An improved map of conserved regulatory sites for Saccharomyces cerevisiae. BMC Bioinformatics 2006;7:113.
23. Maetschke SR, Madhamshettiwar PB, Davis MJ, Ragan MA. Supervised, semi-supervised and unsupervised inference of gene regulatory networks. Brief Bioinform. 2014;15(2):195-211.
24. Marbach D, Costello JC, Küffner R, Vega NM, Prill RJ, Camacho DM, Allison KR, Kellis M, Collins JJ, Stolovitzky G. Wisdom of crowds for robust gene network inference. Nature Methods 2012;9(8):796-804.
25. Marbach D, Prill RJ, Schaffter T, Mattiussi C, Floreano D, Stolovitzky G. Revealing strengths and weaknesses of methods for gene network inference. Proc. Natl. Acad. Sci. U.S.A. 2010;107(14):6286-91. Marbach D, Schaffter T, Mattiussi C, Floreano D. Generating realistic in silico gene networks for performance assessment of reverse engineering methods. Journal of Computational Biology 2009;16(2):229-39.
26. Mitchell TM. Machine Learning. Burr Ridge, IL: McGraw Hill, 1997. Mordelet F, Vert JP. SIRENE: supervised inference of regulatory networks. Bioinformatics 2008;24(16):i76-82.
27. Orlando DA, Lin CY, Bernard A, Wang JY, Socolar JES, Iversen ES, Hartemink AJ, Haase SB. Global control of cell-cycle transcription by coupled CDK and network oscillators. Nature 2008;453(7197):944-7.
28 .Pe'er D, Hacohen N. Principles and strategies for developing network models in cancer. Cell 2011;144(6):864-73.
29. Prill RJ, Marbach D, Saez-Rodriguez J, Sorger PK, Alexopoulos LG, Xue X, Clarke ND, Altan- Bonnet G, Stolovitzky G. Towards a rigorous assessment of systems biology models: the DREAM3 challenges. PLoS One 2010;5(2):e9202.
30. Salgado H, Peralta-Gil M, Gama-Castro S, Santos-Zavaleta A, Muniz-Rascado L, Garcia-Sotelo JS, Weiss V, Solano-Lira H, Martinez-Flores I, Medina-Rivera A, Salgado-Osorio G, Alquicira- Hernandez

© 2019 IJSRET

S, Alquicira-Hernandez K, Lopez-Fuentes A, Porron-Sotelo L, Huerta AM, Bonavides-Martinez C, Balderas-Martinez YI, Pannier L, Olvera M, Labastida A, Jimenez- Jacinto V, Vega-Alvarado L, Del Moral-Chavez V, Hernandez-Alvarez A, Morett E, Collado- Vides J. RegulonDB v8.0: omics data sets, evolutionary conservation, regulatory phrases, crossvalidated gold standards and more. Nucleic Acids Res. 2013;41(Database issue):D203-13.

31. Shalem O, Dahan O, Levo M, Martinez MR, Furman I, Segal E, Pilpel Y. Transient transcriptional responses to stress are generated by opposing effects of mRNA production and degradation. Mol. Syst. Biol. 2008;4:223.
32. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T. Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res. 2003;13(11):2498-504.
33. Vapnik V. Statistical Learning Theory. New York: Wiley, 1998.
34. Vignes M, Vandel J, Allouche D, Ramadan-Alban N, Cierco-Ayrolles C, Schiex T, Mangin B, De Givry S. Gene regulatory network reconstruction using Bayesian networks, the Dantzig Selector, the Lasso and their meta-analysis. PLoS One 2011;6(12):e29165. Yona AH, Manor YS, Herbst RH, Romano GH, Mitchell A, Kupiec M, Pilpel Y, Dahan O. Chromosomal duplication is a transient evolutionary solution to stress. Proc. Natl. Acad. Sci. U.S.A. 2012;109(51):2101

© 2019 IJSRET