Int. J. Nonlinear Anal. Appl. 12 (2021) No. 2, 145-161
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2021.5026

Predicting with the quantify intensities of transcription factor-target genes binding using random forest technique

Ameer K. AL-Mashanji[a,*], Sura Z.AL-Rashid[a]

[a] University of Babylon, Hilla, Iraq

(Communicated by Madjid Eshaghi Gordji)

Abstract

With the rapid development of technology, this development led to the emergence of microarray technology. It has the effect of studying the levels of gene expression in a way that makes it easier for researchers to observe the expression levels of millions of genes at the same time in a single experiment. Development also helped in the emergence of powerful tools to identify interactions between target genes and regulatory factors. The main aim of this study is to build models to predicate the relationship (Interaction) between Transcription Factors (TFs) proteins and target genes by selecting the subset of important genes (Relevant genes) from original data set. The proposed methodology comprises into three major stages: the genes selection, merge data sets and the prediction stage. The process of reducing the computational space of gene data has been accomplished by using proposed mutual information method for genes selection based on the data of gene expression. In the prediction, the proposed prediction regression techniques are utilized to predict with binding rate between single TF-target gene. It has been compared the efficiency of two different proposed regression techniques including: Linear Regression and Random Forest Regression. Two available data sets have been utilized to achieve the objectives of this study: Gene's expression data of Yeast Cell Cycle data set and Transcription Factors data set. The evaluation of predictions performance has been performed depending on two performance prediction measures (Root_Mean_Square Error_(RMSE) and Mean_Absolute_Error_(MAE) with (10) Folds-Cross Validation.

Keywords: Microarray Technology, Gene Expression, Genes Selection, Prediction Techniques, Transcription Factors Proteins.

*Corresponding author

Email addresses: ameeruobabylon@gmail.com (Ameer K. AL-Mashanji), sura_os@itnet.uobabylon.edu.iq (Sura Z.AL-Rashid)

Received: August 2020 *Accepted:* September 2020

1. Introduction

In every cell of living organism, there are basic hereditary units known as genes. Genes are considered base stone which are indispensable for the basic activities of organisms under certain growth conditions [13]. Genes is segment of DNA (Deoxyribonucleic Acid) that holds the genetic information needed by an organism to develop. DNA strand's contain thousands of genes that made of millions of particles called nucleotides [2]. Genes are coded to the protein in the process called gene expression to carry out the fundamental biological functions. Transcription is the process of transferring genetic information from a portion of DNA to the mRNA (Messenger Ribonucleic Acid) [2]. Several complexes, known as Transcription Factors (TFs) proteins, are required for successful transcription. The process is started when one or more particular (TFs) Proteins bind on the gene promoter region to one (or more) special sequences of nucleotides called the Transcription_Factor_Binding_Site (TFBS) [2, 24, 6]. Gene selection algorithms are one of the important steps to select the most useful genes [17]. Since (TFs) proteins have an important role to play in transcription process through their binding with the target gene. Changes in the activities of these factors affect at the level of gene expression and thus lead to a difference in the biological functions. Identifying these relationships is an important biological problem that many scientists are interested in pursuing [9]. Machine learning techniques become one of the most important an automatic and intelligent learning technique, which has the ability to discover hidden knowledge and build models / hypotheses in huge data sets. It have been successfully implemented in several fields such as the markets and the banks and have achieved great successes [3]. Understanding the mechanism of gene regulatory plays an important role in biomedical research. A common challenge is to infer the binding rate between a single TF-target gene, including non-linearity of relationships among the data of gene expression. Therefore, inference algorithms must be able to capture nonlinear relationships between data. Several machine learning techniques such as Convolutional Neural Networks(CNNs), Support Vectors Machine(SVM) and Random Forest have been applied to predict relationship between single (TF) protein and target gene [3]. Among the different methods that have been developed so far, a random forest has emerged as a powerful player. It is an ensemble learning algorithm, which takes an extensive sampling (bootstrap) and random selection of features and combines output from a set of decision trees to extract the final output. Research interests have doubled in the use these methods and it became more important to analysis biological data in the past years [3].

This paper is structured as follows: The related work has been illustrated in Section 2. Microarray technology and expression matrix has been explained in Section 3. Section 4 demonstrates the materials and methods. Section 5 shows the proposed methodology. The results and discussion are presented in Section 6. Finally, the conclusion is explained in Section 7.

2. Related works

fM. Wang et al. [23] applied (CNNs) deep learning model to predict the TF binding intensities between transcription factor and target genes. The datasets have used in study are GM12878 and K562 datasets. The proposed approach demonstrated the Pearson Correlation Coefficient (PCC) for both datasets as follows (0.754) and (0.793) respectively [12]. Zhongxin et al. [25] used the mutual information method for gene selection from high dimensional data. Genes are ranked depending on the mutual information score in descending order. A gene that has highest mutual information score is then selected and gene that has lowest score is then filtered from gene set. The authors used five public data sets in this work namely, Lung, Leukemia, Prostate and Colon. The results of experimental demonstrate that proposed method has ability to select (100) genes, it has better performance to reduce the high dimension of data and enhancing accuracy [25]. In this work Petralia et al. [15] used machine learning technique namely, random forest to predict the binding of transcription factor and target genes. They used the data of gene expression (Exp) and knockout experiments (KO) data set in their experiments. They found that Random Forests is achieved predictive accuracy (0.657) better than Bayesian model [15]. Cui et al. [7] utilized (SVM) machine learning techniques to predict the TF binding and target genes. The dataset that used in this work Thaliana data set. The result of proposed approach demonstrated the model is achieved accuracy, Precision and F1-score as follows (0.9602), (0.8319) and (0.6288) respectively [7].

3. Microarray technology and expression matrix

A microarray chip is a silicon (or a glass) slide, which one of the good tools that comprises of an array of spots. Microarray technique is able to monitor the levels of expression for thousands of genes by estimating the mRNA amount to each location in microarray at the same time. So in one experiment with a microarray slide with n spots, expression levels of n genes can be studied at the same time. The microarray slide is then excited with a laser at appropriate wavelengths then, the final result is stored as an image file [4]. The images of the microarray are taken and analyzed using special software for image analysis and the resulting in an intensity matrix. The puma package which is a tool to extract the data from raw image file (CEL file)[13]. **Figure 1** describes data extraction from the raw gene expression image file. Every intensity matrix is converted into a vector that represents a column of the gene expression matrix with (n) rows (genes). The experiment is repeated for (m) conditions and the results are stored in a (n × m) gene expression matrix, it is demonstrated in Figure 2. Rows within the matrix express the genes expression level and columns express the special sample(conditions). The (ij)th entry of the expression matrix expresses the expression level of ith gene under j^{th} condition [12].

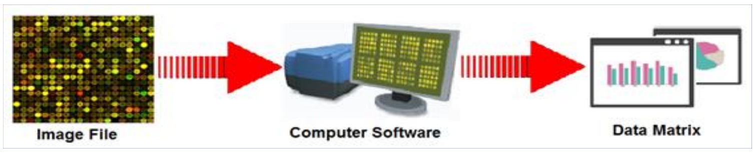


Figure 1: The process of extract data from image file

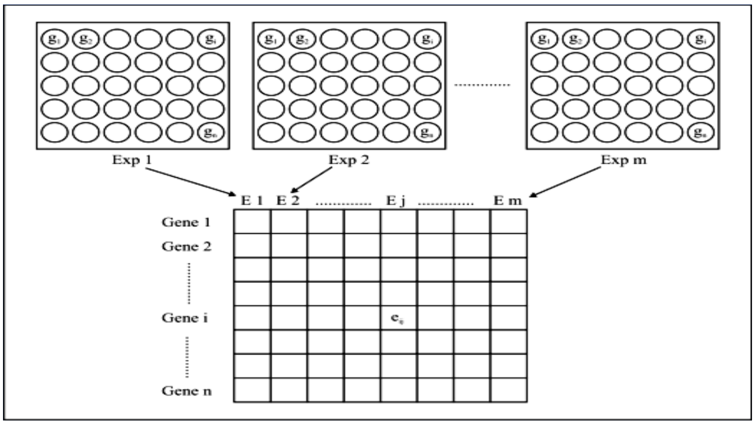


Figure 2: Gene's expression Matrix, each (column, row) corresponds to a (condition, gene), respectively.

4. Materials and methods

4.1. Data sets

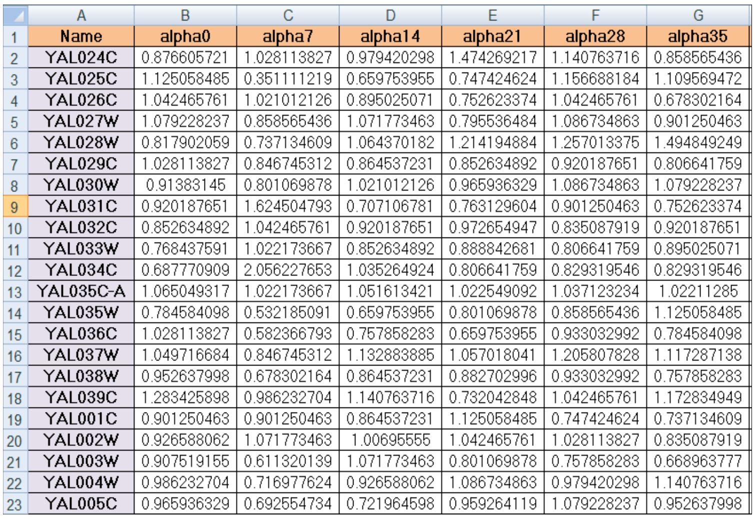
Two available data sets have been used to achieve the objective of this study. The detailed description of the used data sets is reviewed as follows:

4.1.1. Description of Yeast cell cycle data set

It is the first data set that has been used in this study. It is generated by the projects' of Spielman and another group of researchers. They were monitoring the gene expression levels of yeast cell genes by using Microarray's technology. The data set involves more than (6000) genes with (78) features (Conditions). Only three experiments have used in this study, which are alpha experiment (Contains 18 conditions), cdc18 experiment (Contains 24 conditions) and cdc28 experiment (Contains 17 conditions). The first column contains an identifier for each row (gene). Each column (Condition) has a label that is in the first row, which describe the time at which the sample was taken on the set of genes. The remaining values in the data set represent the genes expression levels, which have been observed for genes under a particular condition [20]. The general information of the data set is summarized in **Table 1** . The data set was downloaded from available source on the website "URL: http://genome- www.stanford.edu/cellcycle/data/rawdata/" in the form of Excel files. A screenshot for the file data in Excel format is depicted in **Figure 3** and the number of conditions in each experiment is illustrated in **Figure** 4.

Table 1: A brief of the Yeast Cell Cycle data set

Title of Database :	Yeast Cell Cycle
Data Set Characteristics :	Multivariate
Attribute Characteristics :	String, Real
Missing Values?	Yes
Number of Instances:	6100
Number of Attributes:	60
Publication:	Spielman et al



	A	B	C	D	E	F	G
1	Name	alpha0	alpha7	alpha14	alpha21	alpha28	alpha35
2	YAL024C	0.876605721	1.028113827	0.979420298	1.474269217	1.140763716	0.858565436
3	YAL025C	1.125058485	0.351111219	0.659753955	0.747424624	1.156688184	1.109569472
4	YAL026C	1.042465761	1.021012126	0.895025071	0.752623374	1.042465761	0.678302164
5	YAL027W	1.079228237	0.858565436	1.071773463	0.795536484	1.086734863	0.901250463
6	YAL028W	0.817902059	0.737134609	1.064370182	1.214194884	1.257013375	1.494849249
7	YAL029C	1.028113827	0.846745312	0.864537231	0.852634892	0.920187651	0.806641759
8	YAL030W	0.91383145	0.801069878	1.021012126	0.965936329	1.086734863	1.079228237
9	YAL031C	0.920187651	1.624504793	0.707106781	0.763129604	0.901250463	0.752623374
10	YAL032C	0.852634892	1.042465761	0.920187651	0.972654947	0.835087919	0.920187651
11	YAL033W	0.768437591	1.022173667	0.852634892	0.888842681	0.806641759	0.895025071
12	YAL034C	0.687770909	2.056227653	1.035264924	0.806641759	0.829319546	0.829319546
13	YAL035C-A	1.065049317	1.022173667	1.051613421	1.022549092	1.037123234	1.02211285
14	YAL035W	0.784584098	0.532185091	0.659753955	0.801069878	0.858565436	1.125058485
15	YAL036C	1.028113827	0.582366793	0.757858283	0.659753955	0.933032992	0.784584098
16	YAL037W	1.049716684	0.846745312	1.132883885	1.057018041	1.205807828	1.117287138
17	YAL038W	0.952637998	0.678302164	0.864537231	0.882702996	0.933032992	0.757858283
18	YAL039C	1.283425898	0.986232704	1.140763716	0.732042848	1.042465761	1.172834949
19	YAL001C	0.901250463	0.901250463	0.864537231	1.125058485	0.747424624	0.737134609
20	YAL002W	0.926588062	1.071773463	1.00695555	1.042465761	1.028113827	0.835087919
21	YAL003W	0.907519155	0.611320139	1.071773463	0.801069878	0.757858283	0.668963777
22	YAL004W	0.986232704	0.716977624	0.926588062	1.086734863	0.979420298	1.140763716
23	YAL005C	0.965936329	0.692554734	0.721964598	0.959264119	1.079228237	0.952637998

Figure 3: The values of Yeast Cell Cycle data set

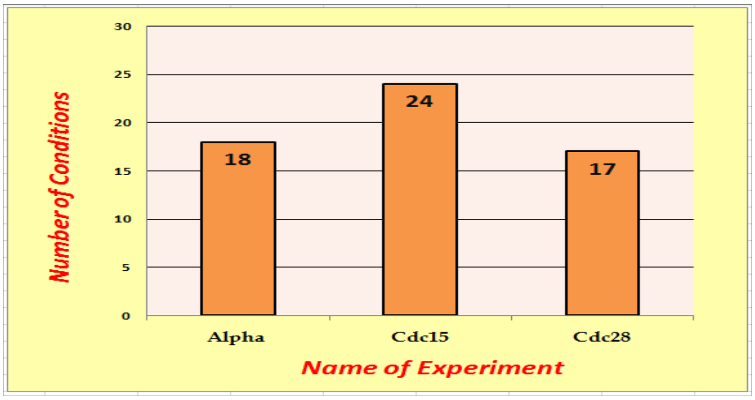


Figure 4: The values of Yeast Cell Cycle data set

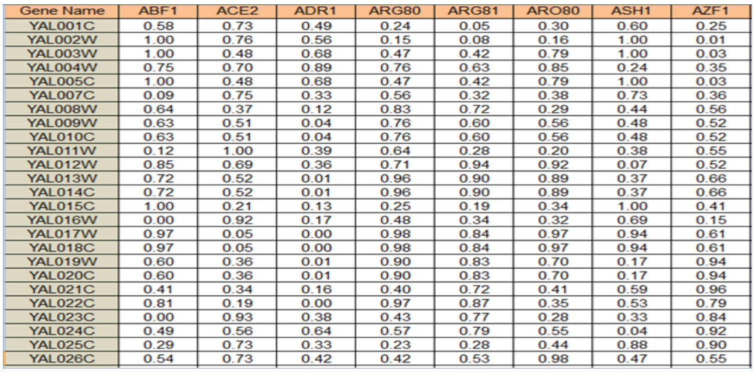
4.1.2. Description of transcription factors data set

It is the second data set that has been used in this thesis. Lee and the researchers analyzed and studied the regulators (transcription factors) that correspond to the genomic regions of the yeast cell genes (data generated by the Spielman project). They utilized chromatin immune-precipitation techniques in their experiments to find the interactions between TF-gene and provide P-values. This data set contains the genes (as rows) with (112) Transcription Factors as features. The first column in data set includes the same identifier of genes, that are exactly in the first data set. Columns have a labels that are appear in the first row, which represent the names of (TFs). The remaining values in the data set represent the (corresponding P-value), which are the binding value between transcription factor (regulator) and the promoter region of the target gene [10]. The information of the data set used **Table 2** and **Figure 5** depicts a screenshot for the file data in Excel format.

Table 2: A brief of the (TFs) proteins data set

Title of Database :	Transcription Factors (TFs)
Data Set Characteristics :	Multivariate
Attribute Characteristics :	String, Real
Missing Values?	No
Number of Instances:	6100
Number of Attributes:	113
Publication:	Lee et al

The data set was downloaded from available source on the web site ”URL: younglab.wi.mit.edu/cgi-bin/young_public/navframe.cgi?s=17&f=downloaddata” in the form of Excel files.



Gene Name	ABF1	ACE2	ADR1	ARG80	ARG81	ARO80	ASH1	AZF1
YAL001C	0.58	0.73	0.49	0.24	0.05	0.30	0.60	0.25
YAL002W	1.00	0.76	0.56	0.15	0.08	0.16	1.00	0.01
YAL003W	1.00	0.48	0.68	0.47	0.42	0.79	1.00	0.03
YAL004W	0.75	0.70	0.89	0.76	0.63	0.85	0.24	0.35
YAL005C	1.00	0.48	0.68	0.47	0.42	0.79	1.00	0.03
YAL007C	0.09	0.75	0.33	0.56	0.32	0.38	0.73	0.36
YAL008W	0.64	0.37	0.12	0.83	0.72	0.29	0.44	0.56
YAL009W	0.63	0.51	0.04	0.76	0.60	0.56	0.48	0.52
YAL010C	0.63	0.51	0.04	0.76	0.60	0.56	0.48	0.52
YAL011W	0.12	1.00	0.39	0.64	0.28	0.20	0.38	0.55
YAL012W	0.85	0.69	0.36	0.71	0.94	0.92	0.07	0.52
YAL013W	0.72	0.52	0.01	0.96	0.90	0.89	0.37	0.66
YAL014C	0.72	0.52	0.01	0.96	0.90	0.89	0.37	0.66
YAL015C	1.00	0.21	0.13	0.25	0.19	0.34	1.00	0.41
YAL016W	0.00	0.92	0.17	0.48	0.34	0.32	0.69	0.15
YAL017W	0.97	0.05	0.00	0.98	0.84	0.97	0.94	0.61
YAL018C	0.97	0.05	0.00	0.98	0.84	0.97	0.94	0.61
YAL019W	0.60	0.36	0.01	0.90	0.83	0.70	0.17	0.94
YAL020C	0.60	0.36	0.01	0.90	0.83	0.70	0.17	0.94
YAL021C	0.41	0.34	0.16	0.40	0.72	0.41	0.59	0.96
YAL022C	0.81	0.19	0.00	0.97	0.87	0.35	0.53	0.79
YAL023C	0.00	0.93	0.38	0.43	0.77	0.28	0.33	0.84
YAL024C	0.49	0.56	0.64	0.57	0.79	0.55	0.04	0.92
YAL025C	0.29	0.73	0.33	0.23	0.28	0.44	0.88	0.90
YAL026C	0.54	0.73	0.42	0.42	0.53	0.98	0.47	0.55

Figure 5: A screenshot of the values of (TFs) proteins dataset

4.2. Data Preprocessing

Data processing methods are an important methods to prepare most data sets in a useful and efficient format. These techniques involve data cleaning tasks such as handling missing values, duplicate data, inconsistent data, and noise removal [1]. Data pre-processing techniques are utilized to converting the original (raw) data into an understandable format, as well as adapting the data to fit the analysis methods that used in subsequent tasks. General databases are often inconsistent, incomplete, and contain many errors. Therefore, pre-processing of data is an important step in solving such problems [19].

4.2.1. Handling missing value

massive amounts of biological data are produced by using microarray technique. The data are usually characterized by an important proportion of missing values .Some causes of missing values as Corruption of image, Irregular spot, Dust and scratches in image, Low intensity, Insufficient resolution, Saturation. And/or Spot variance. Some cases of missing values for gene expression image file are demonstrate visually in **Figure 6** [17].

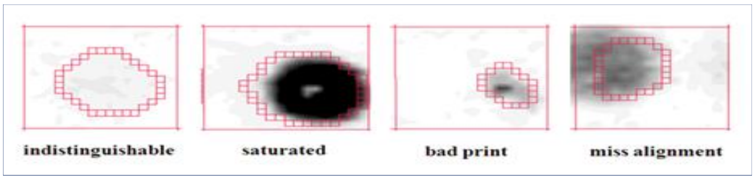


Figure 6: Some Cases of Missing Values

Missing values in the data set can impact the performance of the clustering and prediction models. There are many various methods which handling missing values ”Eliminate Data Objects”, ”Ignore the Missing Value During Analysis” and ”Estimate Missing Values”

The missing values estimation is the most significant type used. In the case of numeric data can use mean method (missing values is replaced by mean value), K-nearest neighbors (KNN) method and others [22]. Mean method has been utilized and the following equation is used to estimate the missing value.

$$Mean = \frac{\sum_{i=1}^{N} X_i}{N} \tag{4.1}$$

where $\boldsymbol{X}$: is the data value. $\boldsymbol{N}$: is the number of values in column.

4.2.2. Data Normalization

It is the step of arranging data in data set. It avoids variation with large values that affect results. The main objective of normalization is to ensure that all values in the data set have the same properties and have the same unit of measurement [22]. In this study, Min-max normalization method has been applied. According to this method the following equation is utilized to calculate the normalization value.

$$V_{\text{norm}} = \frac{V - \min_A}{\max_A - \min_A}(new_max_A - new_min_A) + new\_min_A \tag{4.2}$$

where: $\boldsymbol{Max_A}$: The maximum value for any feature. $\boldsymbol{Min_A}$: The minimum value for any feature. New_max $_A$ and new_min $_A$ are a maximum and minimum interval of values. $\boldsymbol{V}$: represents the feature value.

4.3. Gene selection

Generally, millions of genes expression data are produced during microarray experiments. Thus, many genes are considered not useful and sometimes cause some problems in future analysis [3]. The selection of efficient gene can significantly reduce the computational tasks for subsequent processes

4.3.1. Mutual information concept

Mutual Information in information theory is defined as a measure of shared dependency between two random variables, which is one of the most effective Genes Selection methods [8]. It determines the amount of information for a given random variable based on the other random variable. The mutual information concept is derived from that of entropy of a random variable [8], more details see [11].

4.4. Prediction techniques

The prediction task is divided into two steps. Classification techniques are used if the values of target (Y) are discrete, and regression techniques are used if the values of target (Y) are continuous [22]. In the biological data set (Especially Transcription Factors data set) there are binding rates between genes and regulatory factors, so regression techniques are appropriate for such data set. Many machine learning algorithms can be used to predict such targets with a set of conditions.

4.4.1. Random forest

Random Forest is an ensemble method for regression and classification tasks. It is built on the creation of a number of decision trees or group learning methods (decision tree methods). It aggregates predictions provided by multiple decision trees, where each tree is built on the values of an independent set of random vectors (Bootstrapped samples) [5]. The random forest algorithm relies on the decision trees model by creating multiple training data subsets from the original data set, then (k decision trees) are constructed by training those subsets. Finally, a random forest algorithm is structured by combining decision trees. The outputs from trees are averaged to determine the final output instead of relying on individual trees. Each instance in (test data set) is predicted by taking the decision of all the trees models that have been constricted [5]. The main idea of the Random Forest Algorithm is presented in **Figure 7** .

Random forests have some key characteristics such as the random forest technique is suitable for both classification and regression tasks. Random Forest algorithm considers one of the most accurate learning algorithms available for many data sets because it uses more than one decision tree. it works efficiently on large databases [5].

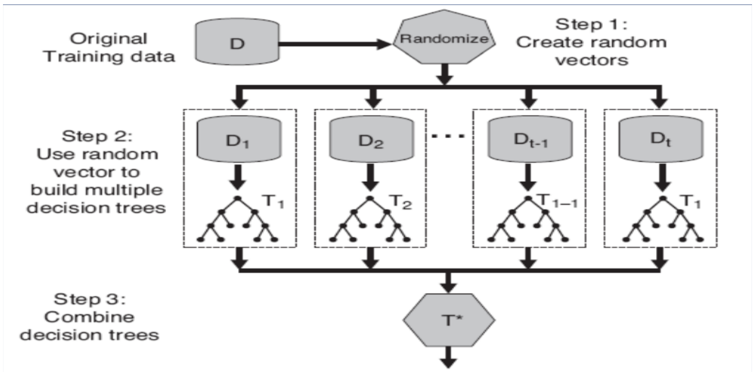

Original Training data
D
Randomize
Step 1: Create random vectors
Step 2: Use random vector to build multiple decision trees
D1
D2
Dt-1
Dt
T1
T2
T1−1
T1
Step 3: Combine decision trees
T*

Figure 7: Random Forest Technique

4.4.2. Linear regression

It is a statistical method that utilized to modeling the relation between a dependent variable (Usually indicated by Y and it means class) and one or more independent variables (or features) using a straight line, more details see [18].

4.5. Evaluation of model performance

There are many common methods to evaluate model performance such as Cross-Validation (C.V), Holdout Method, and others. These methods use the test data set (That is, the data which the model not seen) to evaluate the model performance. C.V method will be used in this study, more details see [16].

4.5.1. Performance metrics

Supervised machine learning (prediction models) has many ways for evaluation machine learning algorithms performance. Prediction models are divided into two kinds : Regression models and classification models .Confusion matrix (e.g. F1 measure ,Precision and Recall) is utilized to measure of the quality of classification model while regression methods are evaluated by computing the error rate (e.g. Mean-Absolute-Error (MAE) and Root-Mean-Squared-Error (RMSE)) [21].

In this study, regression methods have been applied for prediction task, so each of (MAE) and Root Mean Squared Error (RMSE) measures are used to evaluate the prediction error of regression models [26]:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|P_i - A_i| \tag{4.3}$$

where: n : instances number, $\boldsymbol{Pi}$: predicted value for record i, Ai: actual value for record i.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(P_i - A_i)^2} \tag{4.4}$$

where: $\boldsymbol{n}$: number of instances, $\boldsymbol{P}_i$: predicted value of record i, Ai :actual value of record i.

5. Methodology

The proposed Methodology divides into five-stages: data-preprocessing, genes selection, merge data sets, prediction models and evaluation measures. Figure 10 explains the proposed methodology stages block diagram.

Stage 1 **Data Preprocessing:** Preprocessing is intended to prepare data set in an appropriate form to the machine learning technique (Prediction). This stage has been used in this study because Yeast Cell Cycle data set (which has been spotted by Spielman and other group) contains an important proportion of missing values, which occur due to some problems of different biological experiments that generated by using microarray technology such as dust and scratches in image, Irregular spots and etc. In this study, Equation (1) in the Section(4.2.1) of the Mean method has been applied to estimate gene expression data missing values by replacing them with the average of column. In Appendix (A), algorithm (1) illustrates the steps of data-preprocessing stage:

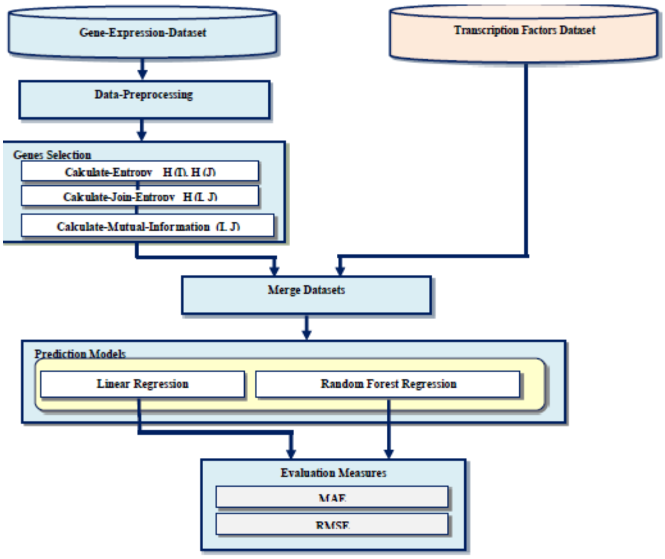
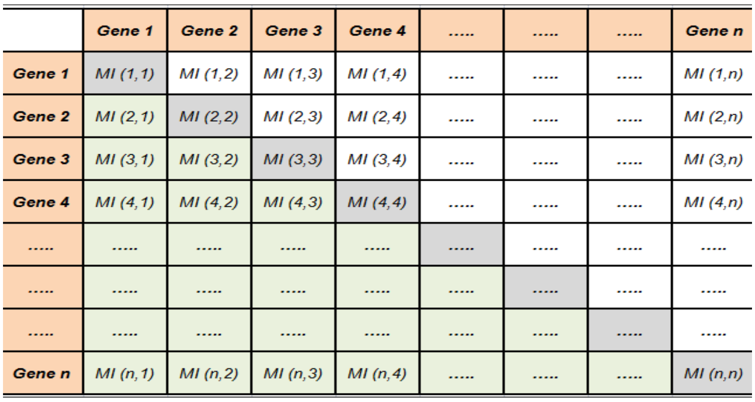


Figure 8: The Architecture of the Proposed Methodology

Stage 2 **Gene Selection :** The main objective of genes selection methods is to reduce the dimensionality of computational space of Yeast Cell Cycle data set. It is the process of selecting genes subset from original genes set because some genes are usually irrelevant. At first, filter methods for genes selection have been used. These methods are always used before the machine learning algorithms and selecting genes based on particular performance measure regardless of the machine learning technique. Mutual information filter method has been used in this thesis. The aim of proposed gene selection method is identifying a subset of genes, which directly used in subsequence tasks (Prediction).In this stage, mutual information has been implemented to identify the important genes and eliminate random genes data depending on the equations in Section (4.2.1). Mutual Information between all genes is calculated in Yeast Cell Cycle data set. The method involves the build of a two-dimensional matrix (n × n), where n expresses genes number in the data set. Figure 11 demonstrates the matrix format of mutual information.

The element at the ith row and jth column is the mutual information between the ith and ith



	Gene 1	Gene 2	Gene 3	Gene 4	.....	.....	.....	Gene n
Gene 1	MI (1,1)	MI (1,2)	MI (1,3)	MI (1,4)	.....	.....	.....	MI (1,n)
Gene 2	MI (2,1)	MI (2,2)	MI (2,3)	MI (2,4)	.....	.....	.....	MI (2,n)
Gene 3	MI (3,1)	MI (3,2)	MI (3,3)	MI (3,4)	.....	.....	.....	MI (3,n)
Gene 4	MI (4,1)	MI (4,2)	MI (4,3)	MI (4,4)	.....	.....	.....	MI (4,n)
.....	.....	.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....	.....	.....
Gene n	MI (n,1)	MI (n,2)	MI (n,3)	MI (n,4)	.....	.....	.....	MI (n,n)

Figure 9: Format of Mutual Information matrix

genes. The main diagonal values of the matrix are substituted with zero values. The values of upper triangle represent the computation of the mutual information between two genes ith and ith. The lower triangle values represent the same values as those of the upper triangle, thus they are replaced by zero values as well. The values of mutual information in upper triangle of the matrix are arranged in descending order ranging from maximum to minimum, and genes that have the highest scores (weights) of mutual information are selected and genes that have less scores are suppressed. These subsets of selected genes are later combined with the (TFs) portions data set, which are passed as input for predication regression models. For additional details about proposed genes selection method looks at the Appendix A, **Algorithm (2)**.

Stages 3 Merge Data sets : At this stage, the subset of the genes (Selected genes) obtained from the previous stage (Genes Selection stage) are combined with the (TFs) portions data set (which has been spotted by Lee and other group). The combination process is done based on the unique gene names in both the subset of genes and (TFs) portions data set. The gene names in the first data set (Selected genes) correspond to the genes expression data, where each gene has a vector of expression levels in different conditions. The gene names in the second data set (TFs) portions correspond to the set of regulatory factors. where each gene corresponds a set of factors with Probability Value (P-Value), which represents binding rate between regulatory factors and target genes. **Figure 12** is the process of merging the two sets of data.

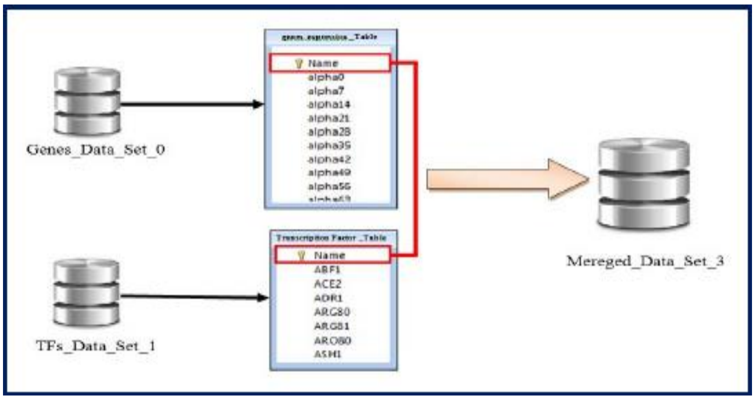
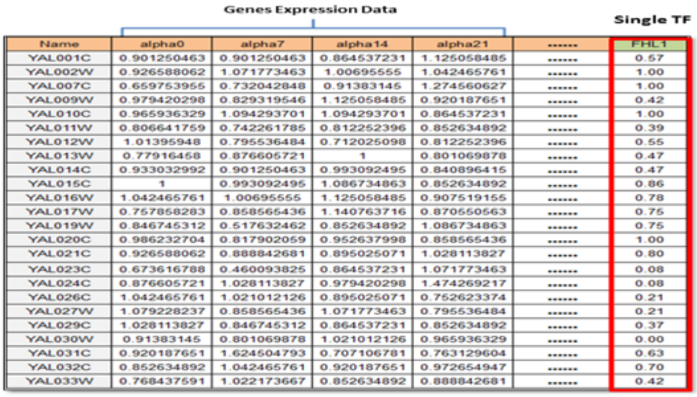


Figure 10: The process of merging datasets

After combining data sets has been done, it can be formed a data set between single transcription factors and genes expression data from the merged data sets process. A single transcription factors Fork Head-Like 1 **(FHLl)** has been used in this study. It is a member of the fork head transcription factors family proteins, which are a group of transcription factors regulating the expression of genes that play important roles in cell development. The formed data set contains unmatched records, which can be used in Prediction task. A sample of (Genes Expression with Single TF) data set after merge as shown in **Figure 13** .



Genes Expression Data						Single TF
Name	alpha0	alpha7	alpha14	alpha21	......	FHL1
YAL001C	0.901250463	0.901250463	0.864537231	1.125058485	......	0.57
YAL002W	0.926588062	1.071773463	1.00695555	1.042465761	......	1.00
YAL007C	0.659753955	0.732042848	0.91383145	1.274560627	......	1.00
YAL009W	0.979420298	0.829319546	1.125058485	0.920187651	......	0.42
YAL010C	0.965936329	1.094293701	1.094293701	0.864537231	......	1.00
YAL011W	0.806641759	0.742261785	0.812252396	0.852634892	......	0.39
YAL012W	1.01395948	0.795536484	0.712025098	0.812252396	......	0.55
YAL013W	0.77916458	0.876605721	1	0.801069878	......	0.47
YAL014C	0.933032992	0.901250463	0.993092495	0.840896415	......	0.47
YAL015C	1	0.993092495	1.086734863	0.852634892	......	0.86
YAL016W	1.042465761	1.00695555	1.125058485	0.907519155	......	0.78
YAL017W	0.757858283	0.858565436	1.140763716	0.870550563	......	0.75
YAL019W	0.846745312	0.517632462	0.852634892	1.086734863	......	0.75
YAL020C	0.986232704	0.817902059	0.952637998	0.858565436	......	1.00
YAL021C	0.926588062	0.888842681	0.895025071	1.028113827	......	0.80
YAL023C	0.673616788	0.460093825	0.864537231	1.071773463	......	0.08
YAL024C	0.876605721	1.028113827	0.979420298	1.474269217	......	0.08
YAL026C	1.042465761	1.021012126	0.895025071	0.752623374	......	0.21
YAL027W	1.079228237	0.858565436	1.071773463	0.795536484	......	0.21
YAL029C	1.028113827	0.846745312	0.864537231	0.852634892	......	0.37
YAL030W	0.91383145	0.801069878	1.021012126	0.965936329	......	0.00
YAL031C	0.920187651	1.624504793	0.707106781	0.763129604	......	0.63
YAL032C	0.852634892	1.042465761	0.920187651	0.972654947	......	0.70
YAL033W	0.768437591	1.022173667	0.852634892	0.888842681	......	0.42

Figure 11: A sample of (Genes Expression with single TF) data set

For additional details about merge data sets step, see Appendix A **Algorithm (3).**

Also in this stage, the data normalization step is applied for genes expression data in order to avoid the difference among the experiments and increase the learning speed of the predication model.

Stage4 Prediction Stage: This stage represents the most important step in the proposed methodology. It has been achieved by implementation the prediction models (Random Forest Regression and Liner Regression). The data set that has been used in this stage is (Genes Expression with Single TF) which has been obtained from the (Merge Data sets Stage). It contains genes expression levels data for each gene and binding rate values of single transcription factor (FHLI). The prediction models are able to predict with the binding value (P-Value) between the a TF (FHL1)-target gene.

Stage5 Evaluation of Prediction Models: In this stage, (RMSE) and (MAE) performance measures have been utilized to measure the prediction error of regression models (Random Forest Regression and Liner Regression Models). Both performance measures have been used to compute predication error rate in data analysis. They measure the difference between two continuous variables (Predicted values and actual values). The concept of (CV) method with (10-Folds) has been used to evaluate the prediction regression models performance to become evaluation that is more reliable. According to this methods each instance is utilized in the same times number for training and exactly once for testing.

6. Results and discussion

The proposed methodology is conducted to demonstrate the effectiveness using mutual information to choose the important genes, examining different prediction regression model's behavior to predicate with binding rate (P-value) between single TF-target genes. At first, gene expression data are read and missing values are processed by applying the mean method (column average). Then, mutual information is computed between two genes in all yeast cell cycle data set. The first (4860) out of (6100) genes identified which have highest mutual information values and are passed to the prediction regression models as input in the next stage. Table 2 present the summary of the reduced data set.

Table 3: A brief information of reduced data set

Title_of_-Dataset :	Yeast- Cell- Cycle
Data_ Set_Characteristics:	Multi-variate
Attribute_ Characteristics :	String,Real
No. of original Instances:	6100
No. of reduced Instances:	4860
No. of Features (Conditions):	60

Then, the subset of the genes (Reduced data set) obtained from the previous stage (Genes Selection stage) are combined with the (TFs) portions data set and formed the data set (Genes Expression with Single TF). In general, Figure 14 explains the level of genes selection method that have been performed to reduce the dimensional of the yeast cell cycle data set and thus identifying the relevant genes with reduction rate.

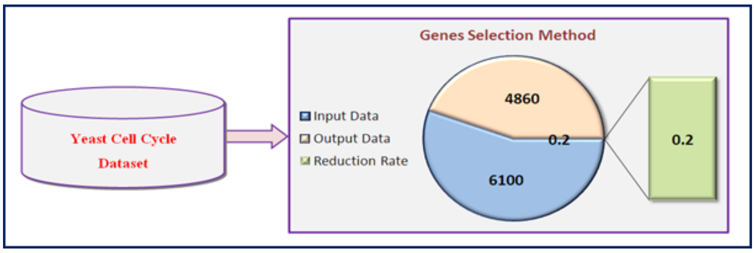


Figure 12: Reduction Level of Gene Selection Method

Finally, random forest regression and linear regression models are implemented based on formed data set. The outcomes of RMSE and MAE for these regression predictors after their implementation to the data set with all features (Conditions) are discussed as follows: The results of MAE and RMSE according to random forest model is summarized in **Tables 3**.

Table 4: Error rate of predication across the Random forest regression

Model	MAE	RMSE
Random Forest Regression	0.182	0.227

Figure 15 shows the bar charts of (RMSE and MAE) of the Random Forest Regression model.

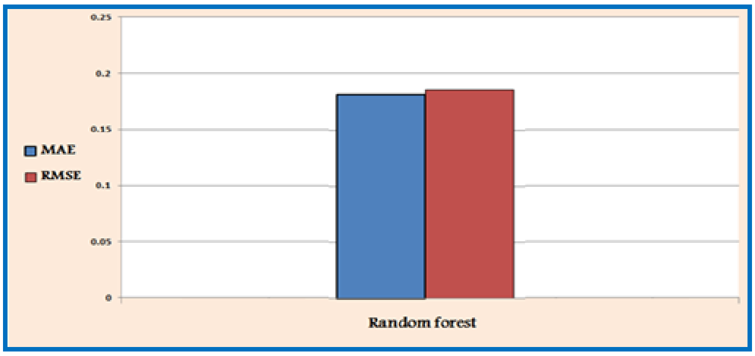


Figure 13: RMSE and MAE of predication across the Random Forest Regression

The results of MAE and RMSE according to linear regression model are summarized in **Table 4**.

Table 5: Error rate of predication across Linear regression Model MAE RMSE Linear Regression

Model	MAE	RMSE
Linear Regression	0.186	0.231

Figure 16 shows the bar charts of RMSE and MAE for the Linear Regression model.

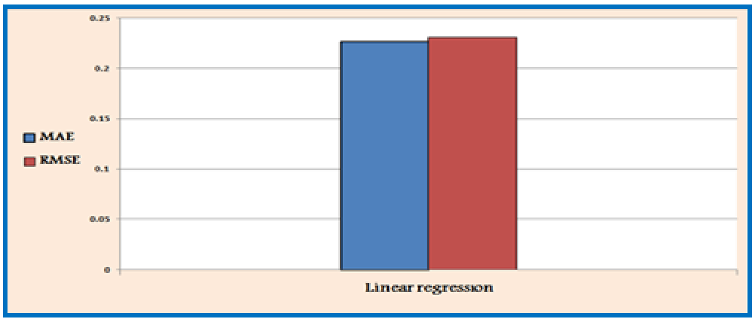


Figure 14: RMSE and MAE of predication across Linear Regression model

The difference between predicted and actual values according to Random Forest model is shown visually in the **Figure 17** while **Figure 18** shows visually the difference between actual and predicted values according to model of Linear Regression.

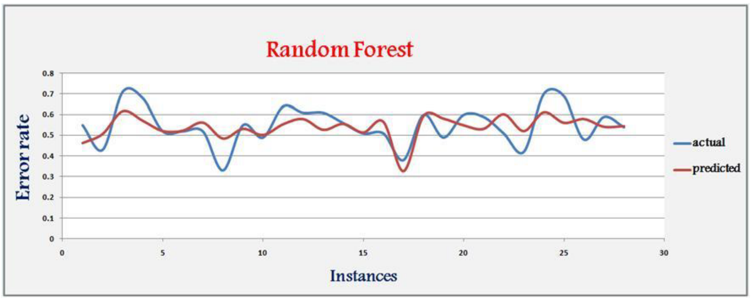


Figure 15: Predicted and actual values with Random Forest Regression -model

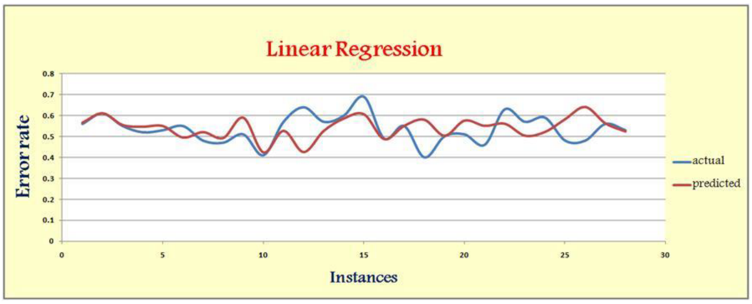


Figure 16: Actual and Predicted values with Linear Regression -model

7. CONCLUSION

This study uses the procedure of mutual information for gene selection to obtain the minimum random gene number and reduce the computational space in order to improve the prediction quality. The proposed prediction models are able to predict the binding rate between the selected genes and the regulation factor (FHL1) after the relevant genes have been identified through using the method of gene selection. The results of prediction regression models indicate that the model of random forest gives a little better result than linear regression model according to performance measures used (MAE and RMAE). The results of experimental demonstrate that the proposed methodology able to eliminate the irrelevant or noises genes data, and it has effectively improved the predicting TF-target gene regulations performance. It also provides additional functional insights for the prediction of gene regulations.

References

[1] A. S. Alasadi and S. W. Bhaya. *"Review of data preprocessing techniques in data mining*, J. Engin. Appl. Sci. 12(16) (2017) 4102—4107.

[2] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts and P. Walter, *Molecular Biology of the Cell*, 4th Edn, New York: Garland Science, 2002.

[3] H. Abusamra, *A comparative study of feature selection and classification methods for gene expression data*, Procedia Comput. Sci. 23 (2013) 5–14.

[4] M. M. Babu, *Introduction to microarray data analysis*, Comput. Genom. Theo. Appl. 17(6) (2004) 225–49.

[5] A. L. Boulesteix, S. Janitza, J. Kruppa and R. Inke König, *"Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics*, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Disc. 2(6) (2012) 493–507.

[6] A. Brazma and J. Vilo, *Gene expression data analysis*, FEBS Lett. 480(1) (2000) 17-–24.

[7] S. Cui, E. Youn, J. Lee and S. J. Maas, *An improved systematic approach to predicting transcription factor target genes using support vector machine*, Phys. Med. 9(4) (2014) ID: 16917899.

[8] K. Das, Kaberi, J. Ray and D. Mishra, *Gene selection using information theory and statistical approach*, Indian J. Sci. Tech. 8(8) (2015) 695—701.

[9] H. Kazan, *Modeling gene regulation in liver hepatocellular carcinoma with random forests*, BioMed Res. Int. 2016 (2016) Article ID 1035945.

[10] T. I. Lee and et al. *Transcriptional regulatory networks in saccharomyces cerevisiae*, Sci. 298(5594) (2002) 799–804.

[11] X. Liu, A. Krishnan and A. Mondry, *An entropy-based gene selection method for cancer classification using microarray data*, BMC Bioinfo. 6(1) (2005) N. 76.

[12] W. Meng, C. Tai, E. Weinan and L. Wei, *DeFine: Deep convolutional neural networks accurately quantify intensities of transcription factor-DNA binding and facilitate evaluation of functional non-coding variants,* Nuc. Acids Res. 46(11) (2018) 69–69.

[13] M. Niklas and G. Mariana, *Definition of Historical Models of Gene Function and Their Relation to Students*, Understanding of Genetics, 2007.

[14] R. D. Pearson, X. Liu, G. Sanguinetti, M. Milo, N. D. Lawrence and M. Rattray, *Puma: A bioconductor package for propagating uncertainty in microarray analysis*, BMC Bioinfo. 10(1) (2009) N. 211.

[15] F. Petralia, P. Wang, J. Yang and Tu. Zhidong, *Integrative random forest for gene regulatory network inference*, Bioinfo. 31(12) (2015) 197—205.

[16] P. Refaeilzadeh, L. Tang and H. Liu, *Cross-validation. Encyclopedia of Database Systems*, (2009) 532–538.

[17] F. Rafii, M. A. Kbir and B. D. R. Hassani, *Microarray data preprocessing to improve exploration on biological databases,* Int. Conf. on Big Data, Cloud and Applications, Tetuan, Morocco, 2015, pp. 25-–26.

[18] S. Slater, S. Joksimovic, V. Kovanovic, B. Vitomir, S. Ryan and D. Gasevic, *Tools for educational data mining: A Review*, J. Educ. Behav. Stat. 42(1) (2017) 85–106.

[19] T. Schlitt and P. Kemmeren, *From microarray data to results*, EMBO Rep. 5(5) (2004) 459-–463.

[20] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein and B. Futcher, *Comprehensive identification of cell cycle–regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization*, 9(12) (1998) 3259–3578.

[21] M. Sokolova, N. Japkowicz and S. Szpakowicz. *Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation*, Aust Joint Conf. Artif. Intel. Springer, 2006 pp. 1015-1021.

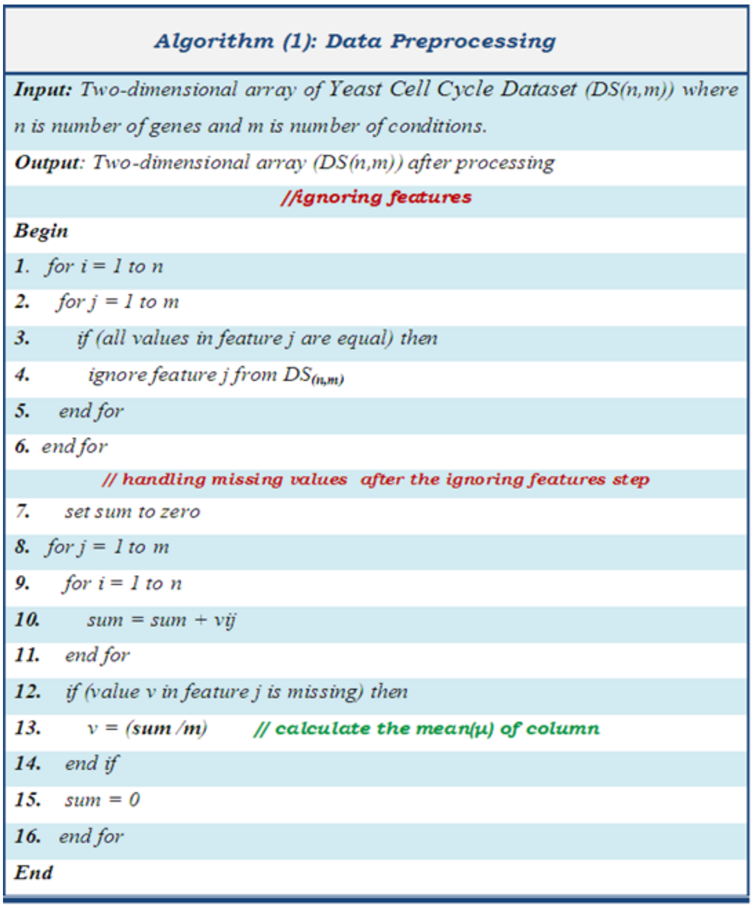
[22] P.-N. Tan, M. Steinbach and V. Kumar. *Introduction to Data Mining*, Pearson Education India, 2006.

[23] W. Wang and Lu. Yanmin, *Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model*, IOP Conference Series: Materials Science and Engineering, IOP Publishing, 2018, 12049.

[24] C. C. Xiang and Y. Chen. *cDNA microarray technology and its applications*, Biotech. Adv. 18(1) (2000) 35–46.

[25] W. Zhongxin, S. Gang, Z. Jing and Z. Jia, *Feature selection algorithm based on mutual information and Lasso for microarray data*, Open Biotech. J. 10 (2016) 278–286.

Appendix A

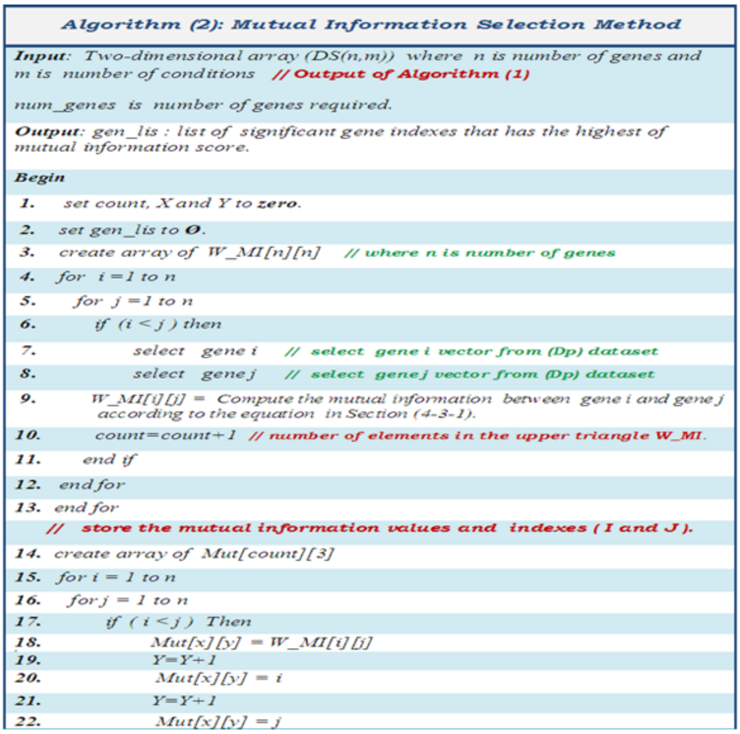


Algorithm (1): Data Preprocessing

Input: Two-dimensional array of Yeast Cell Cycle Dataset (DS(n,m)) where n is number of genes and m is number of conditions.

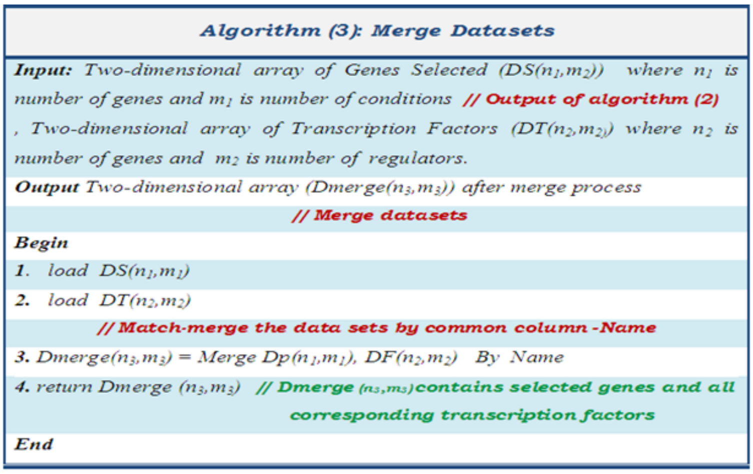
Output: Two-dimensional array (DS(n,m)) after processing

```
                //ignoring features
Begin
  for i = 1 to n
    for j = 1 to m
      if (all values in feature j are equal) then
        ignore feature j from $DS_{(n,m)}$
    end for
  end for
        // handling missing values  after the ignoring features step
    set sum to zero
  for j = 1 to m
    for i = 1 to n
      sum = sum + vij
    end for
    if (value v in feature j is missing) then
      v = (sum /m)        // calculate the mean(μ) of column
    end if
    sum = 0
  end for
End
```



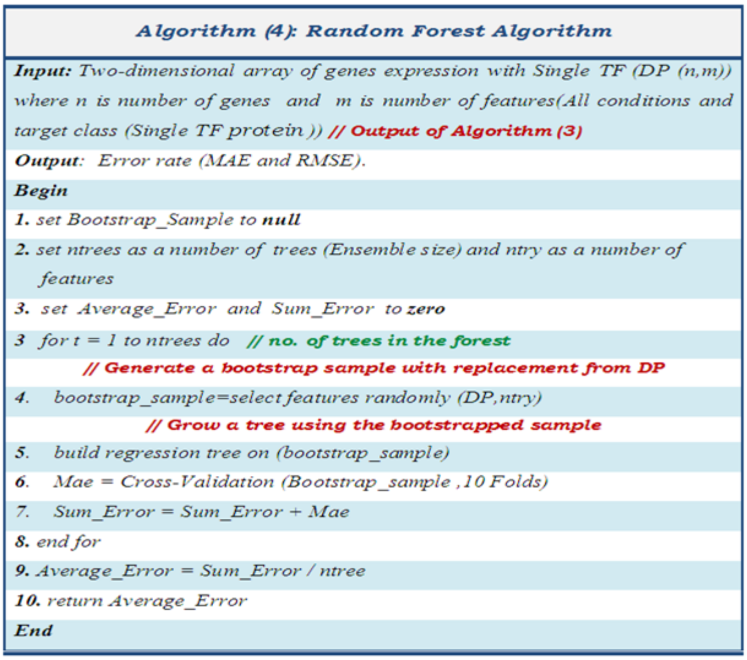
Algorithm (2): Mutual Information Selection Method

```
Input: Two-dimensional array (DS(n,m)) where n is number of genes and
m is number of conditions   // Output of Algorithm (1)

num_genes is number of genes required.

Output: gen_lis : list of significant gene indexes that has the highest of
mutual information score.

Begin
    set count, X and Y to zero.
    set gen_lis to Ø.
    create array of W_MI[n][n]   // where n is number of genes
    for i =1 to n
      for j =1 to n
        if (i < j ) then
            select gene i   // select gene i vector from (Dp) dataset
            select gene j   // select gene j vector from (Dp) dataset
        W_MI[i][j] = Compute the mutual information between gene i and gene j
         according to the equation in Section (4-3-1).
        count=count+1  // number of elements in the upper triangle W_MI.
       end if
    end for
   end for
   //  store the mutual information values and indexes ( I and J ).
   create array of Mut[count][3]
   for i = 1 to n
    for j = 1 to n
       if ( i < j ) Then
           Mut[x][y] = W_MI[i][j]
           Y=Y+1
           Mut[x][y] = i
           Y=Y+1
           Mut[x][y] = j
```



Algorithm (3): Merge Datasets

```
Input: Two-dimensional array of Genes Selected (DS($n_1$,$m_2$)) where $n_1$ is
number of genes and $m_1$ is number of conditions // Output of algorithm (2)
, Two-dimensional array of Transcription Factors (DT($n_2$,$m_2$)) where $n_2$ is
number of genes and $m_2$ is number of regulators.
Output Two-dimensional array (Dmerge($n_3$,$m_3$)) after merge process
                    // Merge datasets
Begin
load DS($n_1$,$m_1$)
load DT($n_2$,$m_2$)
          // Match-merge the data sets by common column -Name
Dmerge($n_3$,$m_3$) = Merge Dp($n_1$,$m_1$), DF($n_2$,$m_2$)  By Name
return Dmerge ($n_3$,$m_3$)  // Dmerge ($n_3$,$m_3$)contains selected genes and all
                                 corresponding transcription factors
End
```



Algorithm (4): Random Forest Algorithm

```
Input: Two-dimensional array of genes expression with Single TF (DP (n,m))
where n is number of genes and m is number of features(All conditions and
target class (Single TF protein )) // Output of Algorithm (3)
Output: Error rate (MAE and RMSE).
Begin
1. set Bootstrap_Sample to null
2. set ntrees as a number of trees (Ensemble size) and ntry as a number of
   features
3. set Average_Error and Sum_Error to zero
3  for t = 1 to ntrees do   // no. of trees in the forest
          // Generate a bootstrap sample with replacement from DP
4.    bootstrap_sample=select features randomly (DP,ntry)
          // Grow a tree using the bootstrapped sample
5.    build regression tree on (bootstrap_sample)
6.    Mae = Cross-Validation (Bootstrap_sample ,10 Folds)
7.    Sum_Error = Sum_Error + Mae
8. end for
9. Average_Error = Sum_Error / ntree
10. return Average_Error
End
```