DOI: 10.24996/ijs.2022.63.11.37

ISSN: 0067-2904

Prediction of DNA Binding Sites Bound to Specific Transcription Factors by the SVM Algorithm

Faisal Abdullah Aziz [*]**, Sura Z. Al-Rashid**

Department of Software, College of Information Technology, University of Babylon, Hilla, Iraq

Received: 7/8/2021 Accepted: 9/2/2022 Published: 30/11/2022

Abstract

In gene regulation, transcription factors (TFs) play a key function. It transmits genetic information from DNA to messenger RNA during the process of DNA transcription. During this step, the transcription factor binds to a segment of the DNA sequence known as Transcription Factor Binding Sites (TFBS). The goal of this study is to build a model that predicts whether or not a DNA binding site attaches to a certain transcription factor (TF). TFs are regulatory molecules that bind to particular sequence motifs in the gene to induce or restrict targeted gene transcription. Two classification methods will be used, which are support vector machine (SVM) and kernel logistic regression (KLR). Moreover, the KLR algorithm depends on another regression algorithm, namely kernel ridge regression (KRR). Discovering binding sites for a transcription factor can help determine genes which it regulates, analyze its functions, comprehend regulation in living organisms, recognize causal disease variations, and, most importantly, manufacture pharmaceutical drugs.

Keywords: SVM; DNA sequences; Transcription Factors (TFs); Kernel Ridge Regression (KRR); Kernel Logistic Regression (KLR).

التنبؤ بمواقع ربط الحمض النووي المرتبطة بعوامل نسخ محددة بواسطة خوارزمية *SVM*

فيصل عبد الله عزيز*, سرى زكي الراشد

قسم البرمجيات, كلية تكنولوجيا المعلومات, جامعة بابل, محافظة بابل, العراق

الخلاصة

في تنظيم الجينات ، تلعب عوامل النسخ (TFs) وظيفة رئيسية. ينقل المعلومات الجينية من الحمض النووي إلى الحمض النووي الريبي المرسال أثناء عملية نسخ الحمض النووي. خلال هذه الخطوة ، يرتبط عامل النسخ بجزء من تسلسل الحمض النووي المعروف باسم مواقع ربط عامل النسخ (TFBS). الهدف من هذه الدراسة هو بناء نموذج يتنبأ بما إذا كان موقع ربط الحمض النووي يرتبط بعامل نسخ معين (TF) أم لا. TFs هي جزيئات تعبيرية ترتبط بأنماط تسلسلية معينة في الجين للحث أو تقييد النسخ الجيني المستهدف. سيتم

[*] Email: faisal.aziz@student.uobabylon.edu.iq

استعمال طريقتين للتصنيف، وهما آلة المتجه الداعمة (SVM) والانحدار اللوجستي للنواة (KLR). علاوة على ذلك ، تعتمد خوارزمية KLR على خوارزمية انحدار أخرى ، وهي انحدار قمة النواة (KRR). يمكن أن يساعد اكتشاف مواقع الربط لعامل النسخ في تحديد الجينات التي ينظمها، وتحليل وظائفها، وفهم التنظيم في الكائنات الحية، والتعرف على الاختلافات المسببة للأمراض، والأهم من ذلك، تصنيع الأدوية الصيدلانية.

1. Introduction

As the machine learning field grows day by day, people are now becoming more and more interested in studying structural data, such as DNA sequences. Techniques for maintaining cell-type-specific gene expression activities are essential in multicellular organisms. The use of chomatin immunoprecipitation followed by sequencing (ChIP-seq) now allows for the genome-wide identification of transcription factors (TFs) and other regulators that regulate these activities [1]. It includes the transformation of genetic information from just a segment of the DNA to a molecule called messenger RNA (Messenger Ribonucleic Acid) [2]. Transcription requires the presence of various complexes, defined as Transcription Factor (TF) proteins. It begins when one or more transcription factors connect to specific sequence sites termed the Transcription Factor Binding Sites (TFBS). Detailed analyses of the TF binding are undoubtedly necessary for future gene expression studies. Many relevant studies have been carried out in the laboratory, such as discovering TF locations and the effect of locus mutations on TF binding. Mutations in TF binding sites and their surrounding regions have a significant impact on gene expression, increasing the risk of complicated disease.

There would be little doubt that precise characterization of TF binding is important for future gene expression research. However, biological TF binding tests are costly and time-consuming. Interdisciplinary bioinformatics is concerned with the gathering, archiving, and arrangement of biological data as well as its analysis and interpretation [3]. Furthermore, it is described as the application of computer techniques for managing biological data in a different declaration. By using machine learning methods and numerical statistics, it can begin to address further issues in the biological sector. Since DNA, RNA, and proteins are among the types of biological data that have increased in volume in the last twenty years, bioinformatics systems have been developed to analyze this data type [4]. The sequence-specific binding of transcription factors to transcription factor binding sites (TFBSs) is critical for transcriptional regulatory control. High throughput TFBS identification methods, such as ChIP-Chip and ChIP-Seq, identify a region of 100–1000 base pairs (b.p.), whereas the actual TFBS is a short region (typically 9–15 b.p.) inside that region [5].

The aim of this research is to determine if the DNA sequence zone is bound to a certain transcription factor (TF). The utilized dataset consists of 2000 training instances and 1000 test examples, with three different TFs for prediction and classification operations. In our work, DNA sequences can be classified into two categories for a particular TF: bound (1) or unbound (0). Two algorithms will be used for classification, namely support vector machine (SVM) and kernel logistic-ridge regression, because they usually provide accurate results. The SVM approach will attract a lot of attention because it may be employed with multiple kernel types. There are three categories of kernels, which are the linear kernel, the quadratic kernel, and the rbf kernel. During the prediction phase, the performance of each kind will be tested, and the accuracy results for each will be displayed. The output of the proposed system is the accuracy of SVM classification with the k-fold cross validation method.

2. Literature Review

Motif finding has always been a difficult task because it requires precise biophysical frameworks to detect TFBS in DNA sequences [6]. Most efforts have been made to predict regulatory regions in DNA sequences. The Cister approach is one of these methods, produced in 2001 by Frith et al. Using the linear-time Forward–Backward method, Cister does not explicitly predict motif clusters, but instead gives a probability curve reflecting the likelihood that every base pair in the sequence will be within a cluster. Cister pays the amount for choosing a little more sophisticated probabilistic model with more annoyance variables. Comet uses the Viterbi technique to locate motif clusters in linear time, but it does not compute the complete log probability value, instead focusing on the most likely arrangement of motifs within the subsequence. Comet has the benefit of calculating E-values to reflect the statistical significance of its predictions.

Cluster-Buster tackles the issue head on, adopting a linear-time heuristic that tries to return the same cluster predictions as the entire quadratic-time technique. We put Cluster-Buster and an implementation of the quadratic-time technique to the test on a collection of 27 short sequences. The two programs produced the same 19 clusters. As a result, Cluster-Buster looks to be very good at emulating the precise procedure. Remarkable efforts have been made to make the Cluster-Buster Web Server as easy to use as possible. For example, each input form choice is connected to a pop-up help box that summarizes its purpose. The output shows an overall image of the motif clusters and labels protein-coding areas in the sequence [7].

3. Methodology

A number of strategies were tried in our search to identify the best preprocessing and methods for our project. In this section, some algorithms that help in the classification process will be explained, such as SVM, regression, k-mer embedding, etc.

3.1 Data Related Experiments

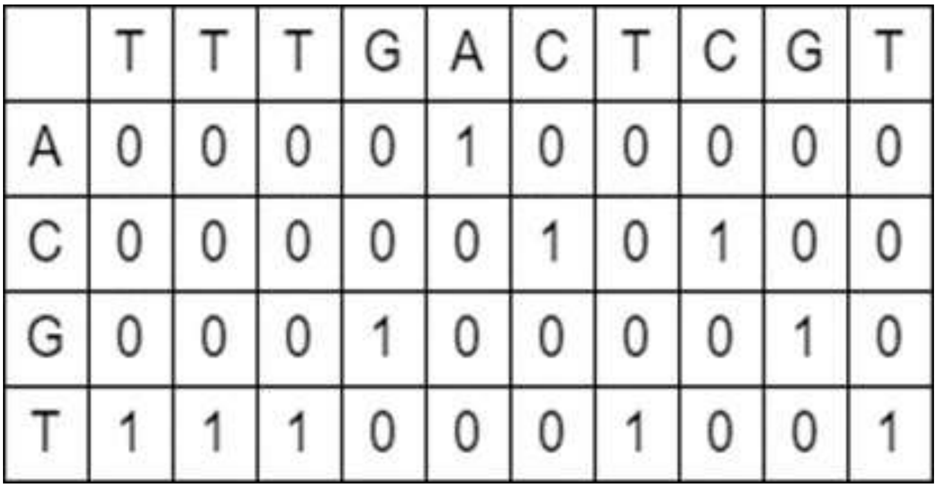
In our machine learning model, numerous approaches are used to encode DNA sequences into normal vectors. 2000 training samples and 1000 test samples were provided. Frequently, DNA sequence data will be in the CSV file format as a table. The software implies that the name of the attribute is being used for the first row of the CSV file.

3.1.1 K-mer Embedding

The first step in data preprocessing is k-mer embedding. This method attempts to encode the co-occurrence information of k-mers into a low-dimensional vector area by utilizing unsupervised learning. Firstly, the input DNA sequence is divided into overlapping k-mers with a specified length k and a stride window s. This results in a k-mer sequence with a size L = $[(l_0\text{-}k)/s] + 1$, with all those k-mers stored in a set C = $[1, 2,\ldots 4^k]$ with a positive integer. For instance, the sequence "ACGGTTAA" will be depicted as [’ACG’,’CGG’, ’GGT’,’GTT’, ’TTA’, ’TAA’]. Then, each K-mer subsequence is mapped into a large vector, and all large vectors are merged into the subsequent space. In addition, how to learn the features of these x ε C^L data sequences with different lengths of L and how to learn a feature map will be studied. Overall, the distributed k-mer representation technique is effective for predicting DNA-binding sites and histones [8].

3.1.2 One-hot Encoding

One-hot encoding is the basic approach for turning DNA sequences into vectors. Thus, the DNA sequence needs to be vectorized into the binary matrix. All DNA sequences must be reduced to a set length during data preprocessing. When the sequence is longer than a specific length, it will be truncated. Use the "N" padding on the specific length if the sequence length is insufficient. In Figure 1, each position might be depicted by a four-case matrix with bases of A, C, G, and T [9].



	T	T	T	G	A	C	T	C	G	T
A	0	0	0	0	1	0	0	0	0	0
C	0	0	0	0	0	1	0	1	0	0
G	0	0	0	1	0	0	0	0	1	0
T	1	1	1	0	0	0	1	0	0	1

Figure 1: Example of one hot encoding on DNA sequences [10]

Thus, a DNA sequence named S = (s1,s2,.....,si,…,sn) with n nucleotides and sequence motif length m, the binary array S for DNA sequence is expressed by a 4×n dimensional column vector as follows:

$$S_{i,j} = \begin{cases} 1 \text{ if } s_{i-m+1} = j^{th}\text{base in } \{A,C,G,T\} \\ 0 \text{ otherwise} \end{cases} \quad (1)$$

j is the index of the column matching to A, C, G, or T, where i is the nucleotide index [8]. Most modern models that work on DNA research utilize the one-hot encoding method to transfer DNA sequences into one-hot arrays and then iteratively upgrade the coefficients of kernel matrices (PWMs) by learning these vectors. The one-hot encoding method is really based on the premise that all nucleotides are statistically independent at bonding sites and participate independently in the appropriate DNA-protein interactions, which makes it challenging to model specific TFBSs. In recent years, several similar studies have shown that considering high-order nucleotide dependence might not only increase discrimination but also provide a better representation of the reasons. Inspired by this finding, a high-order encoding approach will be suggested that takes nucleotide dependency into consideration [6].

3.2 Algorithm Related Experiments

This section will provide clarification on the classification methods utilized in this project, particularly SVM, kernel logistic regression (KLR), and kernel ridge regression (KRR).

3.2.1 Kernel Logistic Regression (KLR)

Logistic Regression (LR) is one of the most significant statistical and data mining approaches for analyzing or classifying binary and proportional data sets used by statisticians and scientists. The approach permits an algorithm to be utilized to classify inbound input based on historical data in a machine learning application. The aim of logistic regression is to assess the likelihood of events, including the relationship between characteristics and the probability of specific results [11]. In the data preparation activity, logistic regression can also play a major role by letting the data sets be placed in specially defined buckets. However, the traditional binary approaches, including LR, are contradictory in respect of unbalanced and infrequent occurrences, as well as limited samples and particular samples (e.g., choice-based sampling), even if specific corrections are used. Preliminary correction and weighting are the most popular correction approaches. Several scientists used these corrections in accordance with the LR method and showed that they can change when the probability of interest in the population is low. KLR (Kernel Logistic Regression) is a robust and adaptable technique of discrimination that offers class prediction confidence. KLR is a logistical regression kernel version (Cawley and Talbot 2008) which transforms the input space into a multi-dimensional kernel functional space. Assume you have a large dataset with n input examples (xi, Li) and xi ∈ R^n, Li ∈ {0,1}. x means the input vector consisting of the fault density, weather, land usage and rainfall, sloping, aspect, altitude, relief amplitude, TWI, SPI, and STi. The two {0,1} classes are landslide and non-landslide. The KLR aims to identify a discriminating feature between land-slide and non-landslide (Eq. (2)) which may divide the two classes [12]. It can be done by means of a logistic regression, which is non-linear:

$$\operatorname{logit}\{y(\mathbf{x})\} = \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{x}) + b \qquad (2)$$

in which w is a model parameter vector and where φ(x) denotes a non-linear input vector conversion. The logit modification of Eq. (2) should be written as:

$$y(\mathbf{x}) = \frac{1}{1 + \exp\{-\mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{x}) - b\}} \qquad (3)$$

In the range [0, 1], this logit function restricts the landslide sensitivity index values of the model. The non-linear transition is called the kernel function between vector representations in the space of the function:

$$\mathcal{K}(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x}) \cdot \boldsymbol{\phi}(\mathbf{x}') \qquad (4)$$

The ideal vector of the model parameter (w), which can be determined by a cost reduction function and represented linearly as follows:

$$\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i \boldsymbol{\phi}(\mathbf{x}_i) \qquad (5)$$

The last figure of logistic regression in the kernel is described as:

$$\operatorname{logit}\{y(\mathbf{x})\} = \sum_{i=1}^{\ell} \alpha_i \mathcal{K}(\mathbf{x}_i, \mathbf{x}) + b \qquad (6)$$

Here the kernel function(xi,x) fulfills the requirements of Mercer (Mercer 1909), b is the intercept function, and α = (α1, α2, ……, αi) is a double-variable model vector.

3.2.2 Kernel Ridge Regression (KRR)

Kernel Ridge Regression (KRR) is a supervised learning process for data analysis and pattern recognition that is also known as LS-SVM (Less square support vector machines) [13]. Kernel Ridge Regression combines Ridge Regression (linear minimum squares) with kernel tricking. The form of the KRR model is the same as that of the supporting vector regression (SVR). While cross-validating LS-SVM or KRR, the training instances are divided into two separate subsets for a given number of times, including one subset of m examples utilized for validation and the other group of (n-m) examples used to build a classifier. Unlike SVR, it is possible to fit a KRR model in a closed format and is usually faster for medium-sized data sets. In recent years, Kernel Ridge Regression (KRR) has become more popular as a complex non-linear data prediction tool that may be applied in several contexts, for example in economics, machine learning, and optical character recognition (OCR), in particular.

For univariate labels, KRR is examined, i.e., a true number is the label y_i. In order to apply RR and KRR to the multi-classification problem of face recognizing, RR and KRR must be extended to the multi-variant label scenario where Y_i is a vector in R^r. The aim of the RR is to find a matrix $W \in R^{p\times r}$, which will model the linear correlation of the x_i with the Y_i label. The classical option is to choose a quadratic cost [14]:

$$J(w) = \sum_i (y_i - w^T x_i)^2 + \lambda \|w\|^2 \tag{7}$$

where λ is a positive value that is stable.

The KRR type is a nonlinear ridge regression where the data sample is substituted by the functional vector: $x_i \longrightarrow \phi_i = \phi(x_i)$ which is derived by the kernel. In fact, the feature vectors cannot be accessed, as is the case with LS-SVM. The new test point x predicted value may be characterized as

$$z_i = \sum_{j=1}^{\ell} \alpha_j \mathcal{K}(\mathbf{x}_j, \mathbf{x}_i) \tag{8}$$

and α can be generated by solving the given linear problem:

$$(K + \lambda I)\alpha = y \tag{9}$$

3.2.3 Support Vector Machine (SVM)

An important classification approach is the supporting vector machine (SVM). The method is based on statistical learning theory. SVMs are one of the strongest predictive algorithms, based on Vapnik (1982, 1995) and Chervonenkis's theory of statistical learning frameworks (1974). In many practical uses, it has shown promising empirical outcomes in man-made and text classification. SVM also operates well with high-dimensional data and prevents the dilemma of dimensionality. Another special property is that it expresses the decision boundary through a portion of the training samples called support vectors. Figure 2 illustrates a graph of a data set containing instances of squares and circles belonging to two separate classes [15].

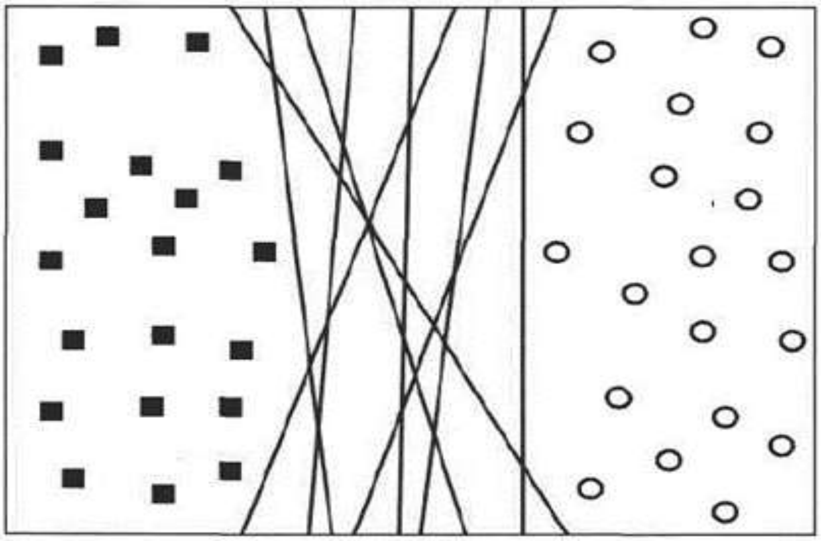


Figure 2: Possible decision boundaries for a linearly separable data set [15]

The SVM approach includes a margin parameter (C) and a kernel variable (sigma) that can deliver the best SVM classification results. Decision limits with high margins are usually more prone to errors than those with narrow margins. If the margin is tiny, every minor disruption to the boundaries of the decision can have a huge effect on its classification. A statistical learning theory defined as structural risk minimization provides a more formal explanation of a linear classifier's limit on its generalization mistakes (SRM). This approach establishes an upper limit to the generalization error in the training error (Re), the number of examples of training (N), and the complexity of a model.

The opposite state to the linear SVM separable case is the linear SVM non-separable case. As Figure 3 shows, these two new examples are misclassified by the decision boundary B_1, but B_2 properly classifies them. This does not imply that B2 is a better decision boundary than B1, because additional examples can be noise in training data. B_1 should still be chosen over B_2, since it has a broader margin and is hence less likely to overfit. But in the preceding part, the SVM wording produces only error-free decision bounds [15].

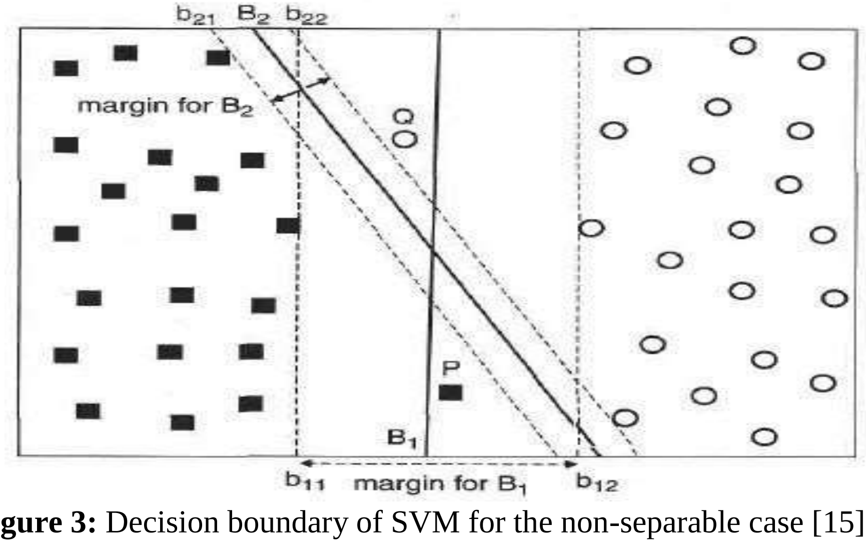


Figure 3: Decision boundary of SVM for the non-separable case [15]

The SVM formulas produce a decision boundary so that the training instances can be separated into your individual classes. There is, however, a method for applying SVM to data sets with non-linear decision bounds. As in the below optimization problem, a non-linear SVM learning process can be structured:

$$\begin{aligned} &\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2} \\ \text{subject to} \quad & y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) \geq 1, \quad i = 1, 2, \ldots, N \end{aligned} \tag{10}$$

There is a similarity between nonlinear SVM and linear SVM equations. The principal distinction is that the learning assignment is carried out on the converted attributes ϕ(x) rather than using the original attributes (x). In nonlinear techniques, certain difficult equations include the calculation of the dot (e.g., similarity) between pairs of vectors, $\phi(x_i) \times \phi(x_j)$. Such computation can be fairly complicated, and the curse of dimensionality can be affected. This difficulty is solved in the form of a process called the kernel trick. The dot product is generally seen as a similarity metric between two input vectors. The dot $\phi(x_i) \times \phi(x_j)$ can likewise be taken as a function of similarity in transformed space between two instances, x_i and x_j. The kernel trick is a way to calculate the similarity of the converted space to the original attribute set. A kernel function K may be represented as

$$K(\boldsymbol{u}, \boldsymbol{v}) = \Phi(\boldsymbol{u}) \cdot \Phi(\boldsymbol{v}) \tag{11}$$

3.3 Kernel Related Experiments

In order to enhance the performance of the utilized classifiers, various kernel kinds that work on string inputs will be investigated and executed. In the following lines, an explanation of the kernels that are used will be offered.

3.3.1 Linear Kernel

When the data is linearly separable, the linear kernel is employed, i.e., it may be split using a single line. It is one of the most frequently utilized kernels. As a result of the refined version of the decision function, linear kernel SVMs have grown in popularity for practical uses since they have quicker training and classification speeds while requiring much less storage than non-linear kernels [16]. It is utilized mostly when a wide range of functions are included in a specific dataset. One example is text classification, where there are many features. In this type, the hyperplane can be found to reside on all squares on one side of the hyperplane and to be located on the other side of all the circles. Based on how well they are intended to do with test instances, the classifier must choose one of those hyperplanes to represent its selection boundary. A linear SVM is a classifier that seeks the largest hyperplane and is hence sometimes known as the maximum classifier of the margins. Generally, the distance between these two hyperplanes is called the classification margin. Take the challenge of binary classification consisting of instances of N training. Each instance is marked with a tuple (x_i,y_i) (i = 1, 2,.., N), where x_i = (x_{i1}, x_{i2},.., x_{id})^T matches to the attribute set for the ith example [15]. By agreement, let $y_i \in \{-1,1\}$ indicate its class label. The linear classifier decision boundary can be established as follows:

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{12}$$

where w and b are model parameters.

The two hyperplanes referred to as bi1 and bi2 are connected with each boundary decision (Bi). (bi1) is obtained by extending a parallel to the decision boundary hyperplane to the nearest square (s), whereas bi2 is acquired by extending the hyperplane to the nearest circle (s). In Figure 4, the two-dimensional spaces are square and circular training sets. A solid line is demonstrated on a decision limit that divides the training instances into their classes. Every example along the border of the decision must correspond with Eq. (12).

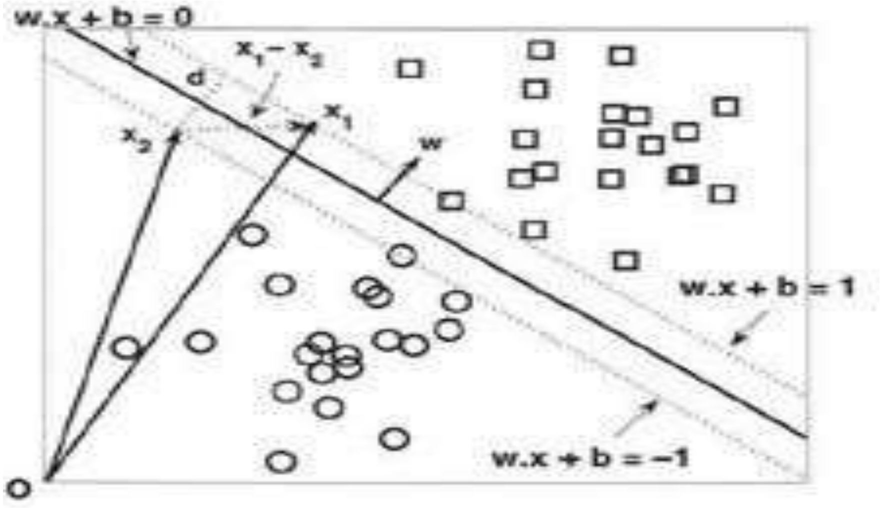


Figure 4: Decision Boundary and margin of SVM [15]

3.3.2 Quadratic Kernel

The quadratic kernel is deemed a special case of the polynomial kernel. The polynomial kernel for machine learning is a mathematical function widely used with SVMs and other kernel-based models that shows the similitude of vectors (training samples) across polynomials of the original variables in a feature space, facilitating the learning of non-linear models. In a non-linear way, this quadratic equation can split data into two classes. Instinctively, the kernel examines not only the characteristics of the input samples but also their combinations. These combinations are known as “interaction features” in the context of regression analysis. The (implicit) functional space of a polynomial kernel is equal to the polynomial regression but not the number of parameters to be learned combined. If the input characteristics are binary (booleans), the characteristics then match the logical conjunctions of the input characteristics. The polynomial kernel is described for degree-d polynomials by:

$$(\square, \square) = (\square^{\square}\square + \square)^{\square} \tag{13}$$

x and y are given input vectors, i.e., vectors of training or test sample computed characteristics, and c ≥ 0 is free of the impact of higher-order and lower-order terms in the polynomial parameter trading. The kernel is known as homogeneous when c = 0. An additional generalized polykernel splits $x^T y$ by a scalar parameter (a) given by the user. Though the RBF kernel in SVM classifying is more prominent than the polynomial, in natural language processing the latter is more popular (NLP). d = 2 (quadratic) is frequent since bigger degrees cause overfit with NLP troubles.

3.3.3 RBF Kernel

The radial basis function kernel (RBF kernel) is a prominent kernel function used in different kernel-linked learning methods. It is utilized especially in support of the classification of vector machines. The most commonly utilized kernel in nonlinear SVM training may be the RBF-Kernel (Gaussian kernel) [17]. This kernel on two examples, x and x', depicted as feature arrays for some input, is defined as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}} \tag{14}$$

$\|\mathbf{x} - \mathbf{x}'\|^2$ describes the Euclidean distance between the two feature vectors, and sigma (σ) is a free RBF kernel variable that controls the kernel weight. The parameters must be modified in SVM to give a more accurate result [18]. The standard value of σ is 1. The gamma parameter (y) utilized in the RBF function is:

$$\gamma = \frac{1}{2\sigma^2} \tag{15}$$

When substituting the value of y in Eq. (14), then it becomes:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2) \tag{16}$$

The RBF kernel is usually an appropriate initial option. The linear kernel is an RBF special state since a linear kernel having a penalty parameter $\tilde{C}$ has some parameters with the same function as the RBF kernel (C, y). Moreover, for specific parameters, the sigmoid kernel acts as an RBF. In contrast to polynomial kernels, the kernelled values of one of the important points are 0 < Kij ≤ 1, which might be infinite to a high degree ($yx^{T} xj + r > 1$) or null ($yx^{T} x + r < 1$). In addition, it must be noted that the kernel sigmoid is not valid under specific variables (i.e., not the internal product of two vectors). Some scenarios are not appropriate for the RBF kernel. In fact, the linear kernel can only be used if the number of features is quite large [19]. Researchers conducted studies and suggest that the SVM (Support Vector Machine) classification technique may be much improved with RBF kernel PSO (Particle swarm optimization) [12].

4. Results and Analysis

The dataset used in this research was obtained from the AGRIS database [20]. Since the data is text, several ways can be used to deal with this data. Our technique is to take a DNA sequence example and divide it into k-mers (the length of a k-mer is equal to 3). Then encode them one-hot to get scarce numerical arrays. The training samples were separated into a training set and a validation set for the training phase. Two classified types have been tested: Kernelized Logistic Ridge Regression (KLR) and Kernelized Soft SVM (KS-SVM). Our experiments were conducted utilizing the Python programming language on a computer with 4GB of RAM, an Intel processor with a Core i5 running at 2.53GHz, and a Windows 10 version. The program used to implement the proposed system is Jupyter Notebook. Cross-validation is used to select the model parameters and hyperparameters. However, selecting a good C (C: SVM parameter), lambda, or sigma parameter was difficult because the outcome was highly dependent on the split between both training and validation data. The following table shows the results of our model:

Table 1: Classification accuracy

Classification Method	Kernel Type	Accuracy %
SVM	rbf kernel	64.5
SVM	quadratic kernel	64.85
SVM	linear kernel	61.501
Logistic-Ridge Regression	linear kernel	63.3

Table 1 shows that the highest accuracy rate achieved was around 65%. This accuracy value was reached by utilizing the SVM algorithm with a quadratic kernel. A series of experiments were carried out to perform a comprehensive comparison of our proposed model with other methods by modifying the database type and classification approach that were used. The newly suggested method produces better results than previous methods. Table 2 highlights the results of earlier studies that were used to determine if a DNA sequence is a binding site for a certain transcription factor (TF) protein. As shown in Figure 5, the FIMO approach was evaluated using the JASPAR.2010 PWMs datasets, which were created via SELEX or single promoter tests [5]. The accuracy of this method was 77.4%. The Patser method recorded 68.7% accuracy, which is less than that of the Mcast method. The PossumSearch tool achieved 67% accuracy, which is higher than that achieved by our method. However, the proposed model achieved a better result than the Comet model, which achieved an accuracy of approximately 63.4%. The Clover achieved a 62.7%, which is higher than the ClusterBuster and Matrix-Scan, with achieved accuracies of 61.7% and 61.2%, respectively. Furthermore, the Cister method achieved an accuracy of 59.9%, which is better than Baycis. The Baycis is considered the worst model among the mentioned models.

Table 2: Comparing between methods performance

Model	**Accuracy**
FIMO	77.4
MCast	72.6
Patser	68.7
PossumSearch	67
proposed model	64.85
Comet	63.4
Clover	62.7
ClusterBuster	61.7
Matrix-Scan	61.2
Cister	59.9
Baycis	54.5

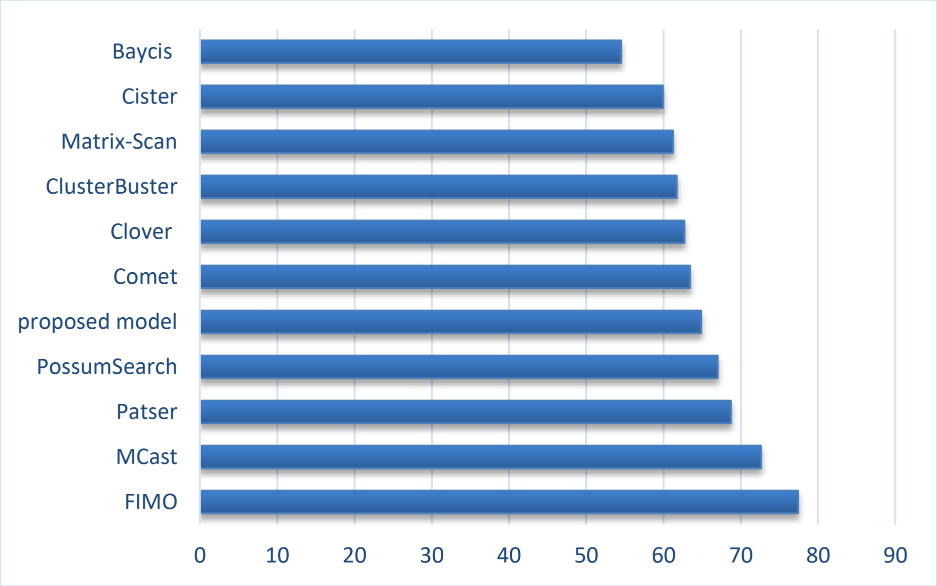


Figure 5: Methodology comparison

5. Conclusions

Several kernels that may be applied to DNA sequences have been implemented and tested throughout this project. To prevent overfitting and obtain satisfying results, a lot of regulation has been applied. In this study, classifiers such as SVM and regression were utilized to predict whether a DNA sequence area is a binding site for a particular transcription factor. Our option of kernel functions may have benefited from a more robust selection method, such as cross-validation. However, this came at the expense of extremely long computational durations, which were difficult to manage during the project due to the restricted computational resources allocated to us. This work, which was both intriguing and exciting, provided a great deal of benefit. Furthermore, some problems discovered during research will be avoided in the future with the tools, information, and experience gained so far.

6. Acknowledgements

I'd like to thank my mentor, Dr. Sura Zaki, for her on-going support and encouragement over the making of this research. I thank my brothers and my family for their assistance and their worthwhile prayers. Furthermore, they assisted me in properly interpreting and analyzing the research content and preparing it.

Conflict of interest

The authors declare that they have no conflicts of interest.

References

[1] A. Arvey, P. Agius, W. S. Noble and C. Leslie, "Sequence and chromatin determinants of cell-type–specific transcription factor binding," *Genome Research*, vol. 22, no. 9, pp. 1723-1734, 2012.

[2] S. Z. AL-Rashid and A. K. Al-Mashanji, "Predicting with the quantify intensities of transcription factor-target genes binding using random forest technique," *International Journal of Nonlinear Analysis and Applications*, vol. 12, no. 2, pp. 145-161, 2021.

[3] S. Z. AL-Rashid and A. K. Al-Mashanji, "Computational Methods for Preprocessing and Classifying Gene Expression Data- Survey," in *4th Scientific International Conference Najaf,* *SICN 2019*, 2019.

[4] N. A. A. Shanan, H. A. Lafta and S. Z. Al-Rashid, "Using alignment-free methods as preprocessing stage to classification whole genomes," *International Journal of Nonlinear Analysis and Applications,* vol. 12, no. 2, pp. 1531-1539, 2021.

[5] N. Jayaram, D. Usvyat and A. C. Martin, "Evaluating tools for transcription factor binding site prediction," *BMC Bioinformatics,* vol. 17, no. 547, pp. 1-12, 2016.

[6] Q. Zhang, L. Zhu and D. S. Huang , "High-order convolutional neural network architecture for predicting DNA-protein binding sites," *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* vol. 16, no. 4, pp. 1184-1192, 2019.

[7] M. C. Frith, Z. Weng and M. C. Li, "Cluster-Buster: Finding dense clusters of motifs in DNA sequences," *Nucleic Acids Research,* vol. 31, no. 13, pp. 3666-3668, 2003.

[8] X. Min, W. Zeng, N. Chen, T. Chen and R. Jiang, "Chromatin accessibility prediction via convolutional long short-term memory networks with k-mer embedding," *Bioinformatics,* vol. 33, no. 14, pp. i92-i101, 2017.

[9] J. Yan and M. Zhu, "A Review about RNA-Protein-Binding Sites Prediction Based on Deep Learning," *IEEE Access,* vol. 8, pp. 150929-150944, 2020.

[10] X. Zhang, B. Beinke, B. A. Kindhi and M. Wiering, "Comparing Machine Learning Algorithms with or without Feature Extraction for DNA Classification," no. November, 2020.

[11] M. Maalouf, "Logistic regression in data analysis: An overview," *International Journal of Data Analysis Techniques and Strategies,* vol. 3, no. 3, pp. 281-299, 2011.

[12] D. Tien Bui, T. A. Tuan, H. Klempe, B. Pradhan and I. Revhaug, "Spatial prediction models for shallow landslide hazards: a comparative assessment of the efficacy of support vector machines, artificial neural networks, kernel logistic regression, and logistic model tree," *Landslides,* vol. 13, no. 2, pp. 361-378, 2016.

[13] J. Verrelst, G. Camps-Valls and F. Veroustraete, "Optical remote sensing and the retrieval of terrestrial vegetation bio-geophysical properties - A review," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 108, pp. 273-290, 2015.

[14] S. An, W. Liu and S. Venkatesh, "Fast cross-validation algorithms for least squares support vector machine and kernel ridge regression," *Pattern Recognition,* vol. 40, no. 8, pp. 2154-2162, 2007.

[15] P. N. TAN , M. STEINBAC and V. KUMAR, Introduction To Data Mining, 1st ed., vol. 13, Boston: Pearson Education, Inc, 2006, pp. 278-296.

[16] S. Maji, A. C. Berg and J. Malik, "Classification using Intersection Kernel Support Vector Machines is Efficient," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[17] Y. W. Chang, C. J. Hsieh and M. Ringgaard, "Training and testing low-degree polynomial data mappings via linear SVM," *Journal of Machine Learning Research,* vol. 11, pp. 1471-1490, 2010.

[18] R. Indraswari, A. Zainal Arifin and D. Herumurti, "RBF KERNEL OPTIMIZATION METHOD WITH PARTICLE SWARM OPTIMIZATION ON SVM USING THE ANALYSIS OF INPUT DATA’S MOVEMENT," *Journal of Computer Science and Information,* vol. 10, no. 1, pp. 36-42, 2017.

[19] V. Apostolidis-Afentoulis and K.-I. Lioufi, "SVM Classification with Linear and RBF kernels," *ResearchGate,* no. July, pp. 1-7, 2015.

[20] A. Yilmaz, M. K. Mejia-Guerra and K. Kurz, "AGRIS: The arabidopsis gene regulatory information server, an update," *Nucleic Acids Research,* vol. 39, no. SUPPL. 1, 2011.