MATEC Web of Conferences **49**, 07001 (2016)
DOI: 10.1051/matecconf/20164907001

Prediction of Bacterial Virulent Proteins with Composition Moment Vector Feature Encoding Method

Murat Gök [1,a], Deniz Herand [2]

[1] *Department of Computer Engineering, Yalova University, Yalova, TURKEY*
[2] *Department of Industrial Engineering, Turkish German University, Istanbul, TURKEY*

Abstract. Prediction of bacterial virulent proteins is critical for vaccine development and understanding of virulence mechanisms in pathogens. For this purpose, a number of feature encoding methods based on sequences and evolutionary information of a given protein have been proposed and applied with some classifier algorithms so far. In this paper, we performed composition moment vector (CMV), which includes information about both composition and position of amino acid in the protein sequence to predict bacterial virulent proteins. The tests were validated in three different independent datasets. Experimental results show that CMV feature encoding method leads to better classification performance in terms of accuracy, sensitivity, f-measure and the Matthews correlation coefficient (MCC) scores on diverse classifiers.

1 Introduction

Virulence is the degree of pathogenicity, which is ability to cause disease, within a group or species of bacteria to invade the tissues of the host. The pathogenicity of an organism is determined by its virulence factors that is typically proteins coded for by genes in chromosomal DNA or plasmids [1]. Identification of virulent proteins that cause drug resistant varieties of various bacterial pathogen evoke the design of peptide based vaccine and drug. Bacterial virulent proteins can be classified from the point of mechanisms of virulence such as adhesion, colonization, invasion, immune response inhibitors and bacterial toxins. Many bacteria adhere to the host cells. This class of proteins includes fimbria and pili in Escherichia coli, Vibrio cholerae, Pseudomonas aeruginosa and Neisseria species [2]. Some virulent bacteria such as helicobacter pylori, which survives in the acidic milieu of the human stomach, produce special proteins that allow them to colonize parts of the host body. Many bacteria produce virulence factors that disrupt the host cell membranes or stimulate their own endocytosis or macro pinocytosis into host cells to allow and facilitate the bacteria to enter host cells [3]. Immune response inhibitors are another class of virulent proteins in some bacteria such as streptococcus pneumoniae that inhibit the host's immune system defenses by producing proteins that bind host antibodies. Bacterial toxins that are made by bacteria that poison host cells and cause tissue damage are another common virulence factors.

Although many bacteria genomes, more than 6000, were sequenced, relatively limited number of virulent proteins were discovered. In the literature, there are two approaches to predict virulent proteins: similarity search and machine learning methods. BLAST [5] and PSI-BLAST [6] are typical examples for the former approach. Machine learning approaches are widely used for the problem. In [7], the authors proposed a neural network-based prediction of virulence factors. Also, 2-gram compositions and the higher order dipeptide composition has been applied with an ensemble of SVM. In [2] the authors have presented a method based on ensemble of classifiers for virulent proteins prediction where the features are extracted directly from the amino acid sequence and from the evolutionary information of a given protein.

The aim of this paper is to apply composition moment vector (CMV), which includes information about both composition and position of amino acid in the protein sequence for prediction of bacterial virulent proteins with several machine learning algorithms.

This paper is organized as follows. In Sect. 2, datasets, feature extraction and classification procedures are explained. In Sect. 3, CMV is evaluated and its performance is obtained according to several classifiers. Finally, Sect. 4 concludes the paper.

2 Material and methods

2.1 Dataset

We conducted our tests on two-to-date datasets: Adhensis (ADH) and Independent (IND) datasets [2, 7]. ADH dataset consists of 469 adhesins and 703 nonadhesins proteins (including several archaebacterial, viral, and

[a] Corresponding author: murat.gok@yalova.edu.tr

yeast non-virulent proteins). IND dataset consists of 367 SWISS-PROT sequences (181 virulent and 186 nonvirulent protein sequences).

2.2 Composition moment vector

Due to the fact that in vitro prediction methods for a protein classification task are time consuming and labour expensive, computational methods based on pattern recognition / machine learning algorithms are used.

A complete pattern recognition system, as shown in Figure 1, consists of a sensor that gathers the observations to be classified, the preprocessing module that removes noise, normalizes the pattern, a feature encoding / selection mechanism that encodes / selects numeric information from the pattern and a classification algorithm that implements the classification task [8].

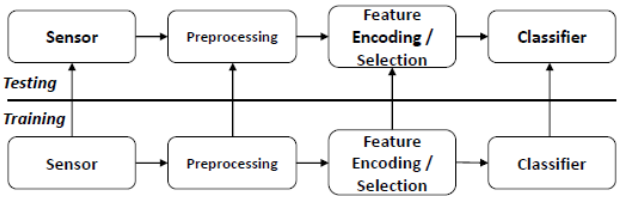

Figure 1. The architecture of a pattern recognition system.

Feature encoding process defines a mapping from the original representation space into a new space where the classes are more easily separable. The goal of feature encoding is to identify the pattern data to the classifier algorithms as much as possible. CMV feature encoding method takes into account amino acid composition and their position information in a sequence. That is, CMV, which includes information about both composition and position of amino acid in the sequence as well as functional relation with the structure content, i.e. there must not be two or more primary amino acid sequences that would have different structure content but the same composition moment vector. Moreover, since it provides information about each AA in the primary sequence, it gives a more comprehensive description of the sequence than other measures [9].

2.3 Classifier algorithms

For classification, we used five types of classifier algorithms: k-Nearest Neighbor (k-NN), Naïve Bayes, Random Forest, C 4.5 and Ada Boost.

The underlying mechanism of k-NN algorithm is quite straightforward that needs no specific training phase. The only things needed are reference data points for both classes (clients, impostors). An unknown (test) data point y is then attributed the same class label as the label of the majority of its k nearest (reference) neighbors. To find these k nearest neighbors the Euclidean distance between the test point and all the reference points is calculated, the obtained distances are ranked in ascending order and the reference points corresponding to the k smallest Euclidean distances are taken. This exhaustive distance calculation step during the test phase leads rapidly to important computing times, which is the major drawback of k-NN [10].

Naïve Bayes is an effective and basic classification algorithm that assumes the feature variables to be independent from each other given the outcome. This assumption simplifies the calculation of conditional probabilities.

With Naïve Bayes algorithm, given a sample, si, the probability of each class, cj, is calculated as in Eq. 1. High probability determined related sample's class.

$$P(c_j \mid s_i) = \frac{P(s_i \mid c_j) \cdot P(c_j)}{P(s_i)} \quad (1)$$

Given a bacterial protein sequence, described by its feature vector $s_i = (s_1, s_2, .., s_n)$, we are looking for a class c_j that maximizes the likelihood $P(s_i \mid c_j) = P(s_1, s_2, .., s_n \mid c_j)$. Thus, each misclassification error, $P(s_i \mid c_j)$, is accounted for s_i with Eq. 1. s_i belongs to the class that provides minimum misclassification error.

The random forest classifier which is a way to improve the performance of Decision Tree consists of a combination of tree classifiers where each classifier is generated using a random vector sampled independently from the input vector, and each tree casts a unit vote for the most popular class to classify an input vector [11]. The random forest classifier used for this study consists of fusing randomly selected features or a combination of features at each node to grow a tree. Design of a decision tree required the choice of an attribute selection measure and a pruning method. There are many approaches to the selection of attributes used for decision tree. Information Gain Ratio criterion [12] and the Gini Index [11] are the main attribute selection measures in decision tree induction. The random forest classifier uses the Gini Index as an attribute selection measure, which measures the impurity of an attribute with respect to the classes [13].

C4.5 algorithm uses a divide-and-conquer approach to learn decision trees which are a very effective method of supervised learning. A decision tree aims the partition of a dataset into groups as homogeneous as possible in terms of the variable to be predicted. It takes as input a set of classified data, and outputs a tree that resembles to an orientation diagram where each end node (leaf) is a decision (a class) and each non- final node (internal) represents a test. Each leaf represents the decision of belonging to a class of data verifying all tests path from the root to the leaf. The data is sorted at every node of the tree in order to determine the best splitting attribute. It uses gain ratio impurity method to evaluate the splitting attribute. Decision trees are built in C4.5 by using a set of training data. At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. Its criterion is the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is chosen to make the decision [14].

The AdaBoost (adaptive boosting) algorithm is an ensemble of classifier algorithms for generating a strong classier out of a set of weak classifiers [15]. In the process of using weak classification repeatedly, different distributions of sample sets should be used, or a weighted sample set, to train the simple classifiers. The weight of a sample set is calculated as in Eq. 2.

$$d^t = \{d_1^t, d_1^t, ..., d_1^t\}, \sum_{n=1}^{N} d_n^t = 1, d_n^t \geq 0 \quad (2)$$

In each training, the misclassified samples will have larger weight during the next training. In general, the samples closest to the decision-making boundary will be easily misclassified. Therefore, after several iterations, these samples assume the greatest weights. As a result, if there are enough training samples and classification errors, we can obtain a stronger classifier through the AdaBoost algorithm [16]. In this study, we have ensemble Decision Stump with AdaBoost classifier.

3 Results and discussion

3.1 Experimental setup

The performances of the classifiers were evaluated by means of accuracy, F-score and the Matthews correlation coefficient (MCC) performance metrics. True positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) values are obtained via confusion matrix. Acc is a widely used measure to determine class discrimination ability, and it is calculated as:

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (3)$$

Specificity is the ratio of TP prediction and it is calculated as:

$$sensitivity = \frac{TP}{TP + FN} \quad (4)$$

Matthew's correlation coefficient (MCC), which is used as a measure of the quality of binary classifications, takes into account true and false positives and negatives.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (5)$$

F-score is a measure of a test's accuracy determining accuracy accounting for both precision and for sensitivity from confusion matrix. F-score accounted as shown in Eq. 6.

$$F - score = 2 \times \frac{precision \times sensitivity}{precision + sensitivity} \quad (6)$$

MCC and F-score takes values between (−1, 1). The higher MCC and F-score get, the more realistic performance classifier system gives [17].

3.2 Performance of CMV

10-fold cross validation (10-fold CV) testing scheme is applied to evaluate the performance of the methods in terms of accuracy, F-score and averaged over 10 experiments on ADH and IND-1 datasets. In a 10-fold CV the training samples are randomly partitioned into 10 equal sized folds with similar class distributions. Each fold in turn is then used as test data for the classifier generated from the remaining nine folds [11]. Having completed the procedures above, the average accuracy, F-score and MCC values of the each method over these 10 turns are obtained, as shown Table 1 and Table 2.

Table 1. Accuracy, F-Score and MCC Performances on ADH Dataset**.**

Classifiers	Accuracy (%)	Sensitivity (%)	F-score	MCC
k-NN	92.66	90	0.90	0.85
Random Forest	91.21	85.90	0.89	0.82
Naive Bayes	85.07	77.60	0.81	0.69
C4.5	84.73	79.50	0.81	0.69
Ada Boost	83.87	72.70	0.78	0.66

The results report that CMV encoding gives the best result with k-NN algorithm with accuracy value of 92.66 %, sensitivity value of 90 % F-score value of 0.93.

Table 2. Accuracy, F-Score and MCC Performances on IND Dataset.

Classifiers	Accuracy (%)	Sensitivity (%)	F-score	MCC
Rotation Forest	71.12	70.70	0.71	0.42
Naive Bayes	69.21	61.30	0.66	0.39
Ada Boost	68.66	65.50	0.66	0.37
C4.5	63.21	54.70	0.60	0.27
k-NN	62.67	58.60	0.61	0.25

The results in Table 2 points out that Rotation Forest algorithm has obtained the best result for accuracy and MCC values with the value of 71.12 % and 0.42, respectively. Note that kNN algorithm obtained the worst performance scores from the point of accuracy and MCC values among learning algorithms.

4 Conclusion

The problem addressed in this paper is to predict bacterial virulent proteins with several machine learning algorithms using CMV feature encoding method on two virulent protein datasets. We performed an experimental comparison of five classifiers: k-NN, Random Forest, Naïve Bayes, C4.5 and Ada Boost. Experimental results

show that CMV method is fit for this problem with different classifiers. Next studies will involve new feature encoding methods which are free of sequence size using ensemble of classifiers. Furthermore, due to the fact that publicly accessible web servers are more practical to study and to develop more useful models or predictors, we will study on developing a web server to predict bacterial virulent proteins.

Acknowledgment

This work was supported by Yalova University, BAP Project (Grant 2015/BAP/103).

References

1. Morens, D. M., Folkers, G. K., and A. S. Fauci, Nature, **430**, 242-249 (2004)
2. Nanni, L., Lumini, A., Gupta, D., and Garg, A. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), **9**, 467-475 (2012)
3. YashRoy, R. C. Indian J Med Res, **126**, 558-566 (2007)
4. Brogden, K. A., Roth, J. A., Stanton, T. B. *et al*. Virulence Mechanisms of Bacterial Pathogens, (third ed.). ASM Press (2000)
5. Altschul, S. F., Gish, W. Miller, W. *et al.* J. Molecular Biology, **215**, 403-410 (1990)
6. Altschul, S. F., Madden, T. L., Schaffer, A. A. *et al*. Nucleic Acids Research, **25**, 3389-3402 (1997)
7. Sachdeva, G., Kumar, K. Jain, P., and Ramachandran, S. Bioinformatics, **21**, 483-91 (2005)
8. Zheng N., Xue J., Statistical Learning and Pattern Analysis for Image and Video Processing, Springer-Verlag, 3-4 (2009)
9. Ruan, J., Wang, K., Yang, J. *et al.* Artificial Intelligence in Medicine, **35**, 19-35 (2005)
10. Verlinde, P., and Cholet, G. Int. Conf. Audio and Video-Based Biometric Person Authentication (AVBPA), 188-193 (1999)
11. Breiman, L. Technical Report 567, Statistics Department, University of California, Berkeley, ftp://ftp.stat.berkeley.edu/pub/users/breiman (1999)
12. Quinlan, J. R., C4.5: Programs for Machine Learning, San Mateo, CA: Morgan Kaufmann (1993)
13. Pal, M. International Journal of Remote Sensing, **26**, 1, 217-222 (2005)
14. Hssina, B, Merbouha, A., Ezzikouri, H., Erritali, M. Int. J. of Adv. Comp. Sci. and App., **4**, 13-19 (2014)
15. Freund, Y. and Shapire, R. Proceedings of the Second European Conference on Computational Learning Theory, 23-37 (1995)
16. Zheng, N. and Xue, J. Statistical learning and pattern analysis for image and video processing, Springer-Verlag London, 172-173 (2009)
17. Gök, M. International Journal of Systems Science, **46**, 1108-1112 (2015)