Meta-Learning for Escherichia Coli Bacteria Patterns Classification

Hafida Bouziane, Belhadri Messabih, and Abdallah Chouarfia

MB University, BP 1505 El M'Naouer
3100 Oran Algeria
e-mail: (h_bouziane,messabih,chouarfia)@univ-usto.dz

Abstract: In machine learning area, there has been a great interest during the past decade to the theory of combining machine learning algorithms. The approaches proposed and implemented become increasingly interesting at the moment when many challenging real-world problems remain difficult to solve, especially those characterized by imbalanced data. Learning with imbalanced datasets is problematic, since the uneven distribution of data influences the behavior of the majority of machine learning algorithms, which often lead to poor performance. It is within this type of data that our study is placed. In this paper, we investigate a meta-learning approach for classifying proteins into their various cellular locations based on their amino acid sequences, A meta-learner system based on k-Nearest Neighbors (k-NN) algorithm as base-classifier, since it has shown good performance in this context as individual classifier and DECORATE as meta-classifier using cross-validation tests for classifying Escherichia Coli bacteria proteins from the amino acid sequence information is evaluated. The paper reports also a comparison against a Decision Tree induction as base-classifier. The experimental results show that the k-NN-based meta-learning model is more efficient than the Decision Tree-based model and the individual k-NN classifier.

Keywords: Classification, Meta-Learning, Imbalanced Data, Subcellular Localization, E.coli.

1. Introduction

Most of the current research projects in bioinformatics deal with structural and functional aspects of genes and proteins. High-throughput genome sequencing techniques have led to an explosion of newly generated protein sequences. Nowadays, the function of a huge number among them is still not known. This challenge provides strong motivation for developing computational methods that can infer the protein's function from the amino acid sequence information. Thus, many automated methods have been developed for predicting protein structural and molecular properties such as domains, active sites, secondary structure, interactions, and localization from only the amino acid sequence information. One helpful step for understanding and therefore, elucidating the biochemical and cellular function of proteins is to identify their subcellular distributions within the cell. Most of the existing predictors for protein localization sites are used with the assumption that each protein in the cell has one, and only one, subcellular location. In each cell compartment, specific proteins ensure specific roles that describe their cellular function which is critical to a cell's survival. This fact means that the knowledge of the compartment or site in which a protein resides allows to infer its function. So far, many methods and systems have been developed to predict protein subcellular locations and one of the most thoroughly studied single cell organism is Escherichia coli (E.coli) bacteria.

The first approach for predicting the localization sites of proteins from their amino acid sequences was a rule based expert system PSORT developed by Nakai and Kanehisa [1,2], then the use of a probabilistic model by Horton and Nakai [3], which could learn its parameters from a set of training data, improved significantly the prediction accuracy. It achieved an accuracy of 81% on E.coli dataset. Later, the use of standard classification algorithms achieved higher prediction accuracy. Among these algorithms, k-Nearest Neighbors (k-NN), binary Decision Tree and Naïve Bayesian classifier. The best accuracy has been achieved by k-NN classifier, that the classification of the E.coli proteins into 8 classes achieved an accuracy of 86% by cross-validation tests [4], The accuracy has been improved significantly compared to that obtained before. Since these works, many systems that support automated prediction of subcellular localization using variety of machine learning techniques have been proposed. With recent progress in this domain, various features of a protein are considered, like composition of amino acids [5], pseudo amino acids [6], and dipeptide and physico-chemical properties [7,8]. The performance of existing methods varies and different prediction accuracies are claimed. Most of them achieve high accuracy for the most populated locations, but are generally less accurate on the locations containing fewer specific proteins. Recently, there has been a great interest to the theory of combining classifiers to improve performance [9]. Several approaches known as ensembles of classifiers (committee approaches) have been proposed and investigated through a variety of artificial and real-world datasets. The main idea behind is that often the ensemble achieves higher performance than each of its individual classifier component. One can distinguish two groups of methods: methods that combine several heterogeneous learning algorithms as base-level classifiers over the same feature set [10], such as stacking, grading and voting, and methods which construct ensembles (homogeneous classifiers) generated by applying a single learning algorithm as base-classifier by sub-sampling the training sets, creating artificial data to construct several learning sets from the original feature set, such as boosting [11], bagging [12] and Random Forests [13]. In protein localization sites prediction problem, data distribution is often imbalanced. For the best of our knowledge, there are two major approaches that try to solve the class imbalance problems: the one which use resampling methods and the one that modify the existing learning algorithms. Resampling strategy balances the classes by adding artificial data for improving the minority class prediction of some classifiers. Here, we focus on the resampling methods, since they are simplest methods to increase the size of the minority class. This article investigates the effectiveness of the meta-learning approach DECORATE |14] to create a meta-level dataset trained using a simple k-NN algorithm as base-classifier in classifying proteins in their subcellular locations in E.coli benchmark dataset using cross-validation and compares the results by using Decision Tree induction as base-classifier.

The rest of the paper is organized as follows. Section 2, presents the materials and the methodology adopted and presents a brief description of E.coli benchmark dataset as well as the evaluation measures used for performance evaluation. Then, section 3 summarizes and discusses the results obtained by the experiments, it also presents a comparison of Decision Tree induction against the k-NN algorithm as base-classifiers to the meta-classifier DECORATE. Finally, section 4 concludes this study.

2. **Material and Methods**

2.1 E.coli Dataset

The prokaryotic gram-negative bacterium Escherichie Coli is an important component of the biosphere, it colonises the lower gut of animals and humans. The Escherichia Coli benchmark dataset has been submitted to the UCI[1] Machine Learning Data Repository [15]. It is well descripted in [1,2,3]. The dataset patterns are characterized by attributes calculated from the amino acid sequences. Protein patterns in the E.coli dataset are classified to eight classes, it is a drastically imbalanced dataset of 336 patterns. One can find classes with more than 130 patterns and other ones with only 2 or 5 patterns. Each pattern with eight attributes (7 predictive and 1 name corresponding to the accenssion number for the SWISSPROT[2] database), where the predictive attributes correspond to the following features : (1) mcg: McGeoch's method for signal sequence recognition [16], the signal sequence is estimated by calculating discriminate score using length of N-terminal positively-charged region (H-region); (2) gvh: Von Heijne's method [17,18] for signal sequence recognition., the score estimating the cleavage signal is evaluated using weight-matrix and the cleavage sites consensus patterns to detect signal-anchor sequences; (3) lip: Von Heijne's Signal Peptidase II consensus sequence score; (4) chg: binary attribute indicating presence of charge on N-terminus of predicted lipoproteins; (5) aac: score of discriminate

[1] Web site: http://archive.ics.uci.edu/ml

[2] Web site: http://www.uniprot.org/

analysis of the amino acid content of outer membrane and periplasmic proteins; (6) alm1: score of the ALOM membrane spanning region prediction program, it determines whether a segment is transmembrane or peripheral; (7) alm2: score of ALOM program after excluding putative cleavable signal regions from the sequence.

Protein patterns in this dataset are organized as follows: 143 patterns of cytoplasm (cp), 77 of inner membrane without signal sequence (im), 52 of periplasm (pp), 35 of inner membrane without uncleavable signal sequence (imU), 20 of outer membrane without lipoprotein (omL), 5 of outer membrane with lipoprotein (omL), 2 of inner membrane without lipoprotein (imL) and 2 patterns of inner membrane with cleavage signal sequence (imS). The class distribution is extremely imbalanced, especially for imL and imS proteins.

2.2 Base-Classifiers

The problem considered here is multi-class, let us denote by Q the number of categories or classes, Q≥3. Each object is represented by its description x ϵ X, where X represents the feature set and its category y ϵ Y, where Y denotes a set of the Q categories and can be identified with the set of indices of the categories: Y={1, …,Q}. The assignation of the descriptions to the categories is performed by means of a classifier, The chosen classifiers are then described in the following subsections.

2.2.1 k-Nearest Neighbors Classifier

The k-nearest neighbors (k-NN) rule [19] is considered as a lazy approach. It is one of the oldest and simplest supervised learning algorithm. Objects are assigned to the class having the majority of the k Nearest Neighbors in the training set. Usually, Euclidean distance is used as the distance metric. Given a test example *x* with unknown class, the algorithm assigns to the example x the class which is most frequent among the *k* training examples nearest to that query example, according to the distance metric. The classification accuracy of k-NN algorithm can be improved significantly if the distance metric is learned with specialized algorithms, many studies try to find the best way to improve the k-NN performance taking into account this factor. In practice, k is usually chosen to be odd. The best choice of this parameter depends on the data concerned with the problem at hand. This algorithm has shown good performance in biological and medical data classification problems.

2.2.2 Decision Tree Induction

A Decision Tree [20] is a powerful way of knowledge representation. The model produced by a decision tree classifier is represented in the form of tree structure. The principle, consists in building decision trees by recursively selecting attributes on which to split. The criterion used for selecting an attribute is information gain. A leaf node indicates the class of the examples.

The instances are classified by sorting them down the tree from the root node to some leaf nodes. Posterior probabilities are estimated by the class frequencies of the training set in each end node. In this study, we used a decision tree built by C4.5 [21].

2.3 Meta-Classifier

Meta-learners such as Boosting, Bagging and Random Forests provide diversity by sub-sampling or re-weighting the existing training examples [14]. Decorate (Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples) performs by adding randomly constructed examples to the training set when building new ensemble members (committee). It has been conceived basing on a diversity measure introduced by the authors. The measure defined expresses the ensemble member disagreement with the ensemble's prediction. If C_j is an ensemble member classifier, $C_j(x)$ the class label predicted by the classifier C_j for the example x and C *(x) the prediction of the ensemble, the diversity d_i of Cj on the example x is defined as follows :

$$d_j(x) = \begin{cases} 0 & if \quad C_j(x) = C^*(x) \\ 1 & otherwise \end{cases} \qquad (1)$$

The diversity of an ensemble of M members, on a training set of N examples is computed as follows :

$$D = \frac{1}{NM}\sum_{j=1}^{M}\sum_{i=1}^{N} d_j(x_i) \qquad (2)$$

The approach consists in constructing an ensemble of classifiers which maximize the diversity measure D. Three parameters are needed: the artificial size which is a fraction of the original training set, the desired number of member classifiers and maximum number of iterations to perform. Initially, the ensemble contains the classifier (base-classifier) trained on the original data. The members added to the ensemble in the successive iteration are trained on the original training data combined with some artificial data. To generate the artificial training examples named as diversity data, the algorithm takes in account the specified fraction of the training set size. The class labels assigned to the diversity data differ maximally from the current predictions of the committee (completely opposite labels). The current classifier is added to the committee if it increases the ensemble diversity, otherwise it is rejected. The process is repeated until the desired committee size is reached or the number of iterations is equal to the maximum fixed. Each classifier C_j of the committee C* provides probabilities for the class membership of each example to classify. If $P_{Cj,k}$ (x) represents the estimated probability of x to belong to the class labeled k according to the classifier C_j, to classify an example x, the algorithm considers the most probable class as the label for x as follows :

$$C^*(x) = \underset{k \in \{1,\ldots,Q\}}{\operatorname{argmax}} P_k(x) \qquad (3)$$

Where P_k (x) represents the probability that x belongs to the class labeled k computed for the entire ensemble , it is expressed as :

$$P_k(x) = \frac{\sum_{C_j \in C^*} P_{C_{j,k}}(x)}{|C^*|} \qquad (4)$$

In this paper, we performed two sets of experiments. In the first one, we used the k-NN classifier as base-classifier. In the second one, we used Decision Tree as base-classifier, which is used in the original DECORATE conception. Our goal was to empirically evaluate the two models on the E.coli dataset. For this purpose, we proceed for the two sets of experiments in two steps. In the first step, we evaluated both the two individual classifiers on Ecoli dataset applying cross-validation and in the second step we used the meta-learning system applying also cross-validation to prediction performance assessment. For all experiments, we made preliminary trials to select the appropriate parameters (model selection).

2.4 Evaluation Measures

Any results obtained by machine learning algorithms must be evaluated before one can have any confidence in their classifications, this aspect of machine learning theory is not only usefull but fondamental. There are several standard methods for evaluation. In what follows, we present only the measures used in this study.

2.4.1 Cross Validation

In this study, we used Cross Validation tests to evaluate the classifier robustness, this methodology is most suitable to avoid biased results. Thus, the whole training set was divided into five mutually exclusive and approximately equal-sized subsets and for each subset used in test, the classifier was trained on the fusion of all the other subsets. So, cross validation was run five times for each classifier and the average value of the five-cross validations was calculated to estimate the overall classification accuracy.

2.4.2 Classification Accuracy Measurements

Some of the most relevant evaluation measures are precision, recall and F-measure. In this study, we adopted the three measures, for evaluating the effectiveness of the classification for each class and the classification accuracy for all the classes as performance measures. A confusion matrix (contingency table of size QxQ has been used, M = $(m_{kl})_{1\leq k,l\leq Q}$, where m_{kl} denotes the number of examples observed in class k and classified in class l. The rows indicate different classes observed and the columns show the result of the classification method for each class. The number of correctly classified examples is the sum of diagonal elements in the matrix, all others are incorrectly classified. The F-measure has two components, which are: the Recall and the Precision. The Recall is the ratio of the number of positive examples (correctly classified) of class k and the number of all positive (observed) examples in class k. We can express this ratio using confusion matrix elements as follows:

$$Recall = 100 \times \frac{m_{kk}}{\sum_{l=1}^{Q} m_{kl}}, k \in \{1, ..., Q\} \qquad (5)$$

The Precision is the ratio of number of correctly classified examples of class k and the number of examples assigned to class k, it can formulated as follows:

$$Precision = 100 \times \frac{m_{kk}}{\sum_{i=1}^{Q} m_{ik}}, k \in \{1, ..., Q\} \qquad (6)$$

The F-measure is then defined as :

$$F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \qquad (7)$$

The classification accuracy is the ratio of number of all correctly classified examples and the total number of examples (both positive and negative), it is given by :

$$Accuracy = 100 \times \frac{1}{N} \sum_{k=1}^{Q} m_{kk} \qquad (8)$$

3. Experimental Results

In this section we report the results for each experiment by highlighting for each step the evaluation measure values. The most important evaluation values are shown with bold typeface. It is important to note, that adding training instance which is common characteristic of DECORATE implies increasing training time. This is visible when performing with a great number of needed classifiers for the ensemble and the desired number of artificial data to create for learning the meta-classifier.

The tables given above report the results of ensembles versus individual classifiers. In this experiment, we applied 5-fold cross-validations. The E.coli dataset is randomly partitioned approximately equally sized subsets. Table 1 and Table 3 summarize the performance in test of each individual classifier for each class. Table 2 and Table 4 give the number of patterns obtained for each class using DECORATE-based k-NN (Dk-NN) and DECORATE-based C4.5 (DC4.5). The best results for k-NN were obtained when setting k=9.

Table 1. Confusion matrix of k-NN as individual classifier on E.coli dataset using 5-CV

Observed	Predicted							
	cp	im	**pp**	**imU**	**om**	**omL**	**imL**	**imS**
cp (143)	**141**	0	2	0	0	0	0	0
im (77)	3	**63**	1	9	0	1	0	0
pp (52)	3	1	**47**	0	1	0	0	0
imU (35)	1	10	0	**23**	0	1	0	0
om (20)	0	0	4	0	**15**	1	0	0
omL (5)	0	0	0	0	0	**5**	0	0
imL (2)	0	1	0	0	0	1	**0**	0
imS (2)	0	1	1	0	0	0	0	**0**

Table 2. Confusion matrix of k-NN based-DECORATE (Dk-NN) on E.coli dataset using 5-CV

Observed	Predicted							
	cp	im	**pp**	**imU**	**om**	**omL**	**imL**	**imS**
cp (143)	**141**	0	2	0	0	0	0	0
im (77)	3	**63**	1	9	0	0	1	0
pp (52)	3	1	**47**	0	1	0	0	0
imU (35)	1	7	0	**26**	0	1	0	0
om (20)	0	0	2	0	**17**	1	0	0
omL (5)	0	0	0	0	0	**5**	0	0
imL (2)	0	1	0	0	0	1	**0**	0
imS (2)	0	1	1	0	0	0	0	**0**

The confusion matrix of Dk-NN in Table 2 shows a gain in classifying om and imU proteins. Whereas, no improvement has been observed for the two minority class proteins namely imL and imS, which are the most difficult to classify.

Table 3. Confusion matrix of C4.5 as individual classifier on E.coli dataset using 5-CV

Observed	Predicted							
	cp	im	**pp**	**imU**	**om**	**omL**	**imL**	**imS**
cp (143)	**137**	2	2	0	2	0	0	0
im (77)	2	**59**	1	13	2	0	0	0
pp (52)	4	2	**45**	0	1	0	0	0
imU (35)	1	12	1	**20**	1	0	0	0
om (20)	1	1	4	0	**14**	0	0	0
omL (5)	0	0	2	0	1	**2**	0	0
imL (2)	0	1	0	0	0	1	**0**	0
imS (2)	0	1	1	0	0	0	0	**0**

Table 4. Confusion matrix of C4.5 based-DECORATE (DC4.5) on E.coli dataset using 5-CV

Observed	Predicted							
	cp	im	**pp**	**imU**	**om**	**omL**	**imL**	**imS**
cp (143)	**142**	0	1	0	0	0	0	0
im (77)	2	**66**	0	8	0	0	0	1
pp (52)	4	2	**46**	0	0	0	0	0
imU (35)	1	13	0	**21**	0	0	0	0
om (20)	0	0	2	0	**18**	0	0	0
omL (5)	0	0	1	0	0	**5**	0	0
imL (2)	0	1	0	0	0	1	**0**	0
imS (2)	0	1	1	0	0	0	0	**0**

Table 3 and Table 5 show that Decision Tree used as individual classifier performs poorly than the individual k-NN. However, in Table 4 the improvement is well observed in both cp, im and om proteins. Not suprisingly, Dk-NN gives better results than DC4.5, which confirms once again its power in this context. What is important to notify is that even the ensembles Dk-NN and DC4.5 fail in classifying pp and imU with high confidence and fail completely for umL and imS. The influence of the number of ensembles (size) needed for the meta-classifier on the performance of the two ensembles Dk-NN and DC4.5 is shown in Fig.1.

Table 5. Test performance using 5- CV on E.coli dataset

Classifiers	**Measures**	**Classes**								**Correctly classified**	**Accuracy**
		cp	***im***	**pp**	**imU**	**om**	**omL**	**imL**	**imS**		
k-NN	**Precision**	95.3	82.9	85.5	71.9	93.8	55.6	0	0	294	**87.5**
	Recall	98.6	81.8	90.4	65.7	75.0	100	0	0		
	F-	96.9	82.4	87.9	68.7	83.3	71.4	0	0		
C4.5	**Precision**	94.5	75.6	80.4	60.6	66.7	66.7	0	0	277	**82.4**
	Recall	95.8	76.6	86.5	57.1	70.0	40.0	0	0		
	F-	95.1	76.1	83.3	58.8	68.3	50.0	0	0		
Dk-NN	**Precision**	95.3	86.3	88.7	74.3	94.4	55.6	0	0	299	**88.9**
	Recall	98.6	81.8	90.4	74.3	85.0	100	0	0		
	F-	96.9	84.0	89.5	74.3	89.5	71.4	0	0		
DC4.5	**Precision**	95.3	79.5	92.0	72.4	100	83.3	0	0	298	**88.6**
	Recall	99.3	85.7	88.5	60.0	90.0	100.0	0	0		
	F-	97.3	82.5	90.2	65.6	94.7	90.9	0	0		

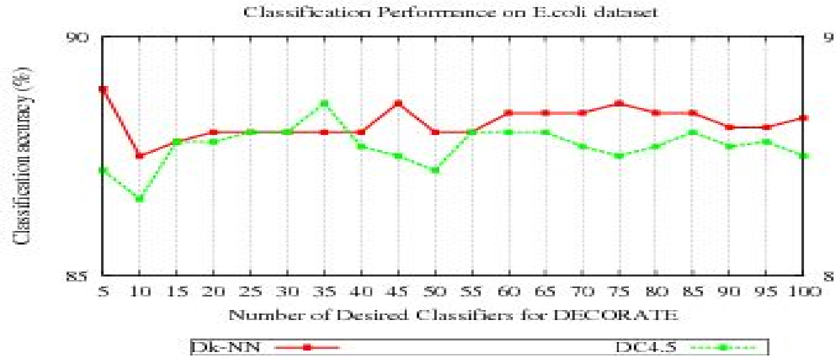

Fig. 1. Comparison of the classification performance (y axis), according to the desired size of classifiers (x axis) between the two ensembles Dk-NN and DC4.5 on E.coli dataset the individual classifiers (x axis)

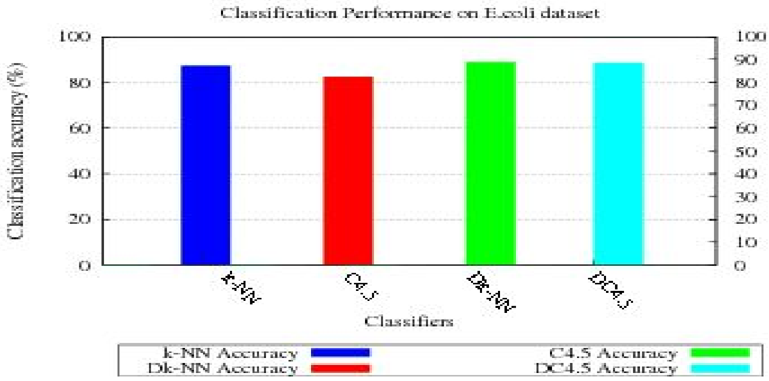

Fig. 2. Accuracy comparison between the four classifiers on E.coli dataset

The results reported for this study show that the classification attempts of inner membrane with lipoprotein (imL) and inner membrane with cleavable signal sequence (imS) proteins failed for each classifier and consequently also for Dk-NN and DC4.5. This situation is caused by the extremely low number of examples in these classes (one example used for training and one example for testing). On the other hand, outer membrane with lipoprotein (omL) proteins were classified with 100% success rate by kNN classifier and both Dk-NN and DC4.5. The cytoplasm (cp) proteins were relatively well classified by almost all classifiers. Fig.2 highlights the performance in test of each classifier and shows well the superiority of the ensembles Dk-NN and DC4.5 in classifying E.coli patterns. Finally; it should be emphasis that this results are better than those obtained by combining heterogeneous classifiers by majority voting rule, since an average classification success of 88.3% was achieved [22]. Nevertheless, all these results prove that combining classifiers is indeed a fruitful strategy.

4. Conclusion

More recently, several ensemble learning algorithms have emerged that have different strengths regardless the type of data involved for the problem in question. One is often confused to make an effective choice among them. Protein cellular localization sites prediction is one among the most challenging problems in modern computational biology. Various approaches have been proposed and applied to solve this problem but the extremely imbalanced distribution of proteins over the cellular locations make the prediction much more difficult. In this study, we applied DECORATE ensemble learning, investigating two standard machine learning approaches to improve the performance in classifying E.coli proteins to their cellular locations, based on their amino acid sequences. The experiments show that the k-NN-based meta-learning model outperforms the individual k-NN classifier and achieves better classification accuracy than the Decision Tree-based model. Further investigations will be carried out to provide a much more improved ensemble model.

5. References

[1] Nakai, K., Kanehisa, M.: Expert system for predicting protein localization sites in gram-negative bacteria. Proteins: Structure, Function, and Genetics. 11,95-110 (1991).

[2] Nakai, K., Kanehisa, M.: A knowledge base for predicting protein localization sites in eukaryotic cells. Genomics. 14, 897-911 (1992).

[3] Horton, P., Nakai, K.: A probabilistic classification system for predicting the cellular localization sites of proteins. In :Proceedings of Intelligent Systems in Molecular Biology, pp 109-115. St. Louis, USA (1996).

[4] Horton, P. , Nakai, K.: Better prediction of protein cellular localization sites with the k Nearest Neighbors classifier, pp. 147-152. AAAI Press. Halkidiki, Greece (1997).

[5] Nakashima, H., Nikishawa, K.: Discrimination of intracellular and extracellular proteins using amino acid composition and residue pair frequencies. J. Mol. Biol. 238, 54–61 (1994).

[6] Park, K. J., Kanehisa, M.: Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. Bioinformatics. 19, 1656–1663 (2003).

[7] Sarda, D., Chua, G.H., Li,K. B., Krishnan, A. :pSLIP: SVM based protein subcellular localization prediction using multiple physicochemical properties. BMC Bioinformatics. 6 ,152 (2005).

[8] Rashid, M., Saha, S., Raghava, G. P. S.: Support Vector Machine-based method for predicting subcellular localization of mycobacterial proteins using evolutionary information and motifs. BMC Bioinformatics. 8, 337 (2007).

[9] Dietterich, T. G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.), First International Workshop on Multiple Classifier Systems, LNCS, pp. 1-15.. Springer-Verlag (2000).

[10] Wolpert, D. H.: Stacked generalization. Neural Networks. 5, 241-259 (1992).

[11] Freund, Y. , Schapire, R. E.: Experiments with a new boosting algorithm. In: Saitta, L. (Ed.), Proceedings of the Thirteenth International Conference on Machine Learning (ICML96). pp. 148-156 (1996).

[12] Breiman, L.: Bagging predictors. Machine Learning. 24 (2), 123-140 (1996).

[13] Rodriguez, J. J., Kuncheva, L. I. : Rotation forest: A new classifier ensemble method. IEEE Transaction in Pattern Analysis. *28*(10), 1619-1630 (2006).

[14] Melville, P. , Mooney, R.: Constructing diverse classifier ensembles using artificial training examples. The Eighteenth International Joint Conference on Artificial Intelligence, pp. 505-510. Acapulco, Mexico, 2003.

[15] Blake, C.L., Merz, C.J.: UCI repository of machine learning databases (1998).

[16] Mcgeoch, D. J., Dolan, A., Donald, S. Rixon, F.J.: Sequence determination and genetic content of the short unique region in the genome of herpes simplex virus type 1. J Mol. Biol. 181, 113 (1997).

[17] Heijne, G. V.: A new method for predicting signal sequence cleavage sites. Nucleic Acids Research. 14, 4683-4690 (1986).

[18] Heijne, G. V. :The structure of signal peptides from bacterial lipoproteins. Protein Engineering. 2,531-534 (1989).

[19] Cover, T. M. Hart, T, P. E.: Nearest neighbor pattern classification. IEEE Transactions on information Theory. 13 (1), 21–27 (1967).

[20] Breiman, L., Friedman, J.H., Olshen, R. A., Stone, C. J.: Classification and regression trees. Monterey, Chapman & Hall (1984).

[21] Quinlan, J.R. :C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo,CA (1993).

[22] Bouziane, H., Messabih, B., Chouarfia, A.: A Voting-Based Combination System for Protein Cellular Localization Sites Prediction. In IEEE International Conference on Information and Computer Applications (ICICA), pp. 166-173, Dubai (2011).