Noname manuscript No.
(will be inserted by the editor)

The impact of pre-Clustering on the Classification of Heterogeneous Protein Data

Haneen Tartouri · Hashem Tamimi · Yaqoub Ashhab

Received: date / Accepted: date

Abstract The aim of this paper is to evaluate the improvement in classification of protein sequence data by introducing clustering as a prepossessing step. We use clustering analysis in order to discover possible sub-clusters that might have different patterns within the same protein class. A classification learning algorithm is then applied to each cluster to enhance the classification accuracy.

Two standard benchmark datasets; Caspase 3 human substrates that include cleaved and non-cleaved peptides, and the membrane proteins inner and /*alpha*-helical proteins were used to examine the proposed approach. Different descriptors based on the physicochemical properties of amino acids were extracted from the protein sequence data and two encoding methods were used to represent the protein sequences using the descriptors.

The results show that applying clustering process prior to classification gives higher prediction accuracy than using classification alone. In addition, the result of time performance shows that the proposed approach succeeded in reducing the training time of the classification process significantly while maintaining the accuracy of prediction.

Keywords Protein sequence data; Classification; Clustering; Physico-chemical properties.

Haneen Tartouri
College of Computer Information Technology and Computer Engineering,
Palestine Polytechnic University,
Hebron Palestine.

Hashem Tamimi (Corresponding author)
College of Computer Information Technology and Computer Engineering,
Palestine Polytechnic University,
Hebron Palestine. E-mail: htamimi@ppu.edu

Yaqoub Ashhab
Palestine-Korea Biotechnology Center,
Palestine Polytechnic University,
Hebron Palestine, E-mail: yashhab@ppu.edu

1 Introduction

Prediction of protein function is an important problem in the area of bioinformatics. It is used to identify the hidden attributes of newly discovered proteins so as to help the researchers to identify functions of unknown proteins in a faster and more cost-effective manner [27].

Different machine learning algorithms are frequently used to classify and predict functional attributes of proteins based on their sequence data only. Support Vector Machine (SVM) [11], random forest [13] and Artificial Neural Network (ANN) [24] are amongst these algorithms. The key step in building a good machine learning classifier is to train the algorithm on data from different classes (functional attributes) so as to be able to classify any new data with unknown class label. The training data has typically a binary nature; one class of the data possess the studied attribute, while the second does not. The more the capability of the classifier to map the training data and separate between the two classes the more accurate it will show in predicting the class of newly unlabeled items. However, protein families are heterogeneous as they represent a long evolutionary history of a wide range of organisms[16]. This heterogeneous nature of protein classes is considered a major challenge to most classification algorithms[5]. It has been noticed that data heterogeneity within the positive or negative classes impedes a good separation between the two classes. In addition, after mapping protein sequence data of several problems, it was found that the data within a given class tends to form various sub-classes, e.g. [4, 29], which is a major hurdle to improve the accuracy of machine learning classifiers. In addition, in the majority of the protein sequence problems, the functional attributes are poorly defined. The sequence feature(s) that dictate a given function are usually hidden in several positions within the protein sequence and there is no white and black rule to find it out.

One of the methods that can be used to solve the above-mentioned challenges is to reduce the heterogeneity of the biological data by grouping the biological datasets based on their similarities. In some cases, researchers try to group the data manually based on empirical analysis in order to increase the accuracy of classification. This method needs time, especially for large datasets, and it can not be applied to all problems. This leads to using computerized algorithms that can group the data based on their features and similarities, such as clustering algorithms. These algorithms divide data into groups based on their features, each group is called a cluster.

In this paper, our aim was to investigate the impact of applying clustering process prior to classification for heterogeneous protein data. Both training time and accuracy of prediction for protein sequences are analyzed by using a combination of classification algorithms and clustering analysis. The aim of the clustering is to map the data into groups based on their similarities in order to reduce the heterogeneity of data, then the classification algorithm is applied to each cluster.

Classification based on clustering has been applied earlier to some studies such as to textual data [15], and random datasets [6, 31] in order to minimize the training time of the classifier, and in some cases to enhance the accuracy of the prediction.

Cervantes, et al. [6] have introduced approaches to reduce the classification time for a large random dataset, they used the fuzzy clustering algorithm and then the SVM algorithm was applied on homogeneous clusters only. This method enabled

them to reduce the training time while maintaining the same range of accuracy [6]. Their approach as we can see eliminates some samples from the dataset, and also it adds an overhead for applying the SVM classifier twice. However, Yu, et al. [31] showed that random sampling could hurt the training process of SVM, especially when the probability distribution of training and testing data were different.

On the other hand, Kyriakopoulou, et al. [15] enhanced the classification result for text data by clustering the data into clusters and then each cluster contributes one meta-feature to the feature space of the training and testing data, finally they used SVM classifier to classify the expanded data , they were able to enhance the classifier results approximately by 8%. The main disadvantage of this method is that the testing data should be involved in the process from the beginning to form the meta-features.

Rahideh, et al. [25] studied the cancer data (colon cancer and leukemia) by using the clustering in order to group the genes and then select the top ranking genes from each group to form the intended subset of relevant genes to be used for classification. As a result, they enhanced the accuracy of the classifiers for the cancer sequences using the clustering algorithm before the classification.

The most important features that distinguish our approach from previous approaches are: there is no need to eliminate samples from the dataset in order to minimize the training time as some previous approaches do [6, 15], and we do not need to involve the new sample in the process from the beginning to form the features.

This approach aims at enhancing the accuracy of the prediction for the protein sequences in the first place which is considered to be more challenging and then examines the effect of our approach on the training time of the classifier.

In this work, different sets of physicochemical properties (PCPs) are used to represent the protein sequences. These properties are divided into two groups; the native properties [30, 26] and the derived properties [28, 12, 10]. The native properties are natural properties taken from the amino acids indices databases, where the derived properties are computed from the native properties using statistical methods as shown later. We have evaluated the performance of the proposed approach on classifying full protein sequences and peptide sequences, specifically in classifying outer membrane proteins and inner membrane proteins, and in classifying cleaved and non-cleaved Caspase 3 peptides.

The remaining parts of the paper are organized as follows: Section 2 introduces the descriptors used in this study. Section 3 presents the encoding methods used in this paper. Section 4 describes the benchmarks of the data used to carry the experiment. Section 5 covers the classification based on clustering methodology used in this study to enhance the performance of the prediction. Section 6 illustrates the steps that used to predict a novel sample. Section 7 demonstrates experiments and the results achieved by the work. Section 8 discusses the results of experiments. Finally, Section 9 and Section 10 are the conclusion and new direction for the future work respectively.

2 Descriptors Used in This Study

In bioinformatics, proteins are represented as strings of characters of variable lengths as follows: Let $s = r_1, r_2, \cdots, r_n$ be a protein sequence of length $|s| = n$

over an alphabet Σ, where r_i represents the i_{th} residue in the sequence, and $\Sigma = \{G, A, V, L, I, P, F, Y, W, S, T, N, Q, C, M, D, E, H, K, R\}$. Each element in Σ is called amino acid. Usually when $n < 50$ we refer to the protein sequence as a *peptide* [19].

The first step in our approach is the selection of the suitable descriptors in order to represent the amino acids of the proteins. In this study, two types of properties are used to examine the performance of the proposed approach; Native properties and derived properties.

The Native properties are the amino acid properties of each amino acid such as its size, hydrophobicity, polarity or inferred propensity. Some existing databases of amino acid indices such as AAindex [14] and APDbase [17] contain hundreds of properties for enriching each amino acid. Some properties in these databases are redundant, Therefore, in the proposed approach, we used a set of non-redundant properties that contains 50 PCPs of amino acids proposed by Georgiev [10].

The derived properties are those properties that were derived from analyzing a large set of PCPs by applying a given reduction algorithm such as Principal Component Analysis (PCA), Multidimensional Scaling (MDS) and Factor Analysis (FA). In the proposed approach we follow the work by Georgive (2009) [10] for deriving the properties as explained in the following section.

3 Encoding Protein Sequences Using PCPs

The encoding methods using PCPs from proteins is the process of representing the protein sequences as numerical features (i.e. feature extraction). The encoding methods also map the variable length protein sequences into fixed length features, which is considered essential for many machine learning classifiers.

The following encoding methods are used in this study:

3.1 Pseudo-Amino Acid Composition

Pseudo-Amino Acid Composition (PseAAC) is very commonly used encoding approach for proteins [20]. It represents a protein sequence with a discrete model without completely losing the amino acid sequence order information. It is formed from weighted sums of amino acid compositions, physicochemical square correlations, combinations of amino acid compositions and dipeptide composition [20]. The feature vector is composed of 20 (from Amino acid Composition (AC)) + λ which is a correlation factor.

If a protein sequence has L amino acids residues: $R_1R_2R_3....R_{L-2}R_{L-1}R_L$ Sequence order effect can be approximately reflected with a set of sequence order-correlated factors as defined below:

$$\begin{aligned}
\theta_1 &= \tfrac{1}{L-1}\textstyle\sum_{i=1}^{L-1}\Theta(R_i, R_{i+1}) \\
\theta_2 &= \tfrac{1}{L-2}\textstyle\sum_{i=1}^{L-2}\Theta(R_i, R_{i+2}) \\
\theta_3 &= \tfrac{1}{L-3}\textstyle\sum_{i=1}^{L-3}\Theta(R_i, R_{i+3}) \\
&\vdots \\
\theta_\lambda &= \tfrac{1}{L-\lambda}\textstyle\sum_{i=1}^{L-\lambda}\Theta(R_i, R_{i+\lambda}) \qquad (\lambda < L)
\end{aligned} \tag{1}$$

The θ_1 is called the first-tier correlation factor that reflects the sequence order correlation between all the most contiguous residues along a protein chain, θ_2 the second-tier correlation factor that reflects the sequence order correlation between all the second most contiguous residues, and θ_λ is the $\lambda - th$ tier correlation factor [7].

The correlation factor can be defined as:

$$\Theta\left(R_i, R_j\right) = \left[F\left(R_j\right) - F\left(R_i\right)\right]^2 \tag{2}$$

where $F(R_i)$ is the feature (e.g. size) value of the amino acid R_i. The value is converted from the original feature value of the amino acid according to the following equation:

$$F\left(R_i\right) = \frac{F_0\left(R_i\right) - \sum_{i=1}^{20} \frac{F_0(R_i)}{20}}{\sqrt{\frac{\sum_{i=1}^{20}\left[F_0(R_i) - \sum_{i=1}^{20} \frac{F_0(R_i)}{20}\right]^2}{20}}} \tag{3}$$

where $F_0(R_i)$ is the original feature value of the amino acid R_i. So, the feature vector (V) of the protein can be represented by a $(20 + \lambda)$ vector as follows:

$$v_x = \begin{cases} \frac{f_x}{\sum_{i=1}^{20} f_i + w \sum_{j=1}^{\lambda} \theta_j} & (1 \leq x \leq 20) \\ \frac{W\theta_{x-20}}{\sum_{i=1}^{20} f_i + w \sum_{j=1}^{\lambda} \theta_j} & (21 \leq x \leq 20 + \lambda) \end{cases} \tag{4}$$

where $f_x (x = 1, 2, \ldots, 20)$ represents the amino acid composition (AC), which was described earlier.

3.2 Composition Transition Distribution encoding

Composition Transition Distribution (CTD) encoding is the famous encoding of proteins that depending on distributing amino acids into groups based on their PCPs [21]. The feature vector is composed of 147 elements, where 21 elements (from Composition) + 21 elements (from transition) + 105 elements (from distribution) for all sequences regardless their lengths.

3.3 Concatenation encoding

In the case of having peptides of fixed length (e.g: Caspase 3) the Concatenation encoding method can be used, this method depending on representing each amino acid numerically as the corresponding set of different physicochemical properties.

4 Datasets

Two datasets of proteins are used to evaluate the performance of the proposed approach on various types of protein features so as to ensure the general applicability of this approach, these dataset are: Membrane proteins dataset is used as a benchmark dataset for full protein sequences, and Caspase 3 is used as benchmark dataset for peptide sequences. To evaluate the performance of the proposed approach, we have hidden part of the data for testing purpose and used the rest for establishing the clusters and the classifiers.

4.1 Membrane proteins dataset

The membrane proteins dataset constructed by Park and coworkers [22] is used to study the performance of the proposed approach for the full protein sequences, it contains 208 outer membrane proteins (OMPs), 673 globular proteins, and 206 α-helical membrane proteins [22]. In our study, we emphasize on identifying the OMPs from inner membrane proteins, so OMPs and the α-helical membrane proteins are selected from the Parks dataset to construct a benchmark contains two classes, where the OMPs represent the positive class and the α-helical membrane proteins represent the negative class.

4.2 Caspase 3 dataset

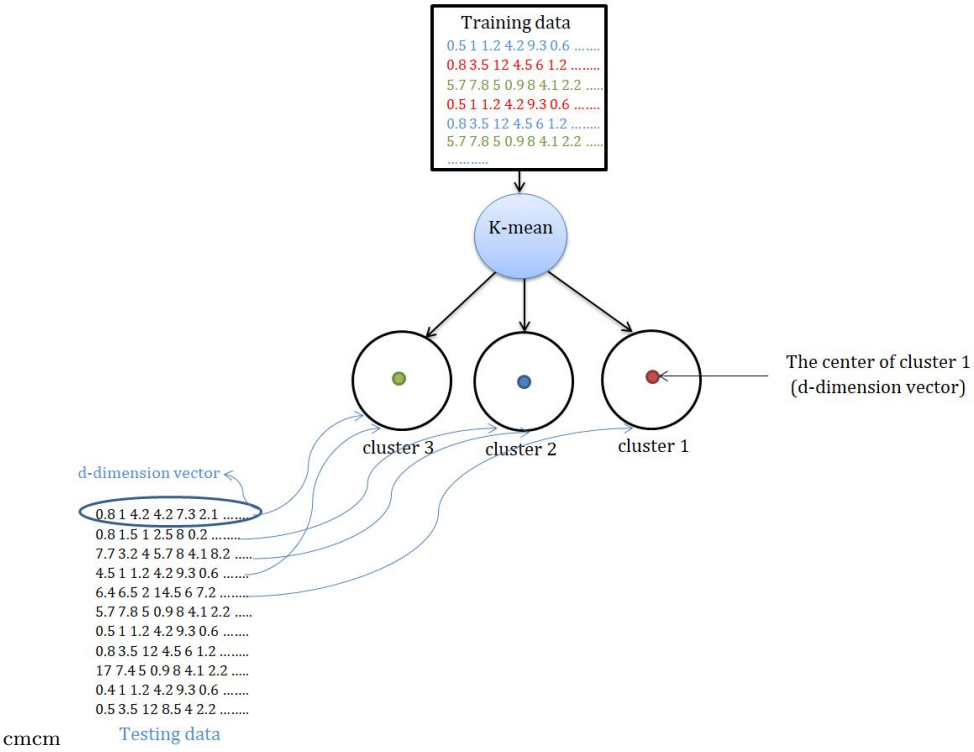
A dataset of Caspase 3 human substrates is used [4], this dataset contains 247 mapped cleavage sites and these sequences represent a positive data. While the negative data are 247 non-cleaved peptides extracted randomly and contained aspartic acid residue 'D' but outside the Caspase 3 cleaved site. The main characteristic of Caspase 3 sequences is that all sequences have the same lengths of 14 amino acids.

5 Classification Based on Clustering

In this study, clustering is used before classification in order to enhance the performance of protein attributes prediction. This section contains a description of the proposed approach in order to explain how clustering can be used before the classification.

In our approach we used the K-mean algorithm to cluster the data, this algorithm was chosen due to its simplicity, and frequent use in the literature [25,15], K represents the number of clusters. Figure 1 illustrates an example of this step, this Figure shows three clusters resulted from applying the clustering algorithm for the data.

Each cluster contains a group of homogeneous data, so we train a classifier for each cluster. In our approach the support vector machine (SVM) was used to classify the data, because it is one of the most powerful classification techniques that was successfully applied to many real-world problems, it has proven a great success in many areas, such as protein classification and face recognition [3], and it is suitable for unbalanced data. See Figure 1.



cmcm

Fig. 1 The proposed approach. The K-mean algorithm is used to group the data, then the SVM classifier is used for each cluster

6 Prediction of a Novel Sample

After training the SVM the prediction of new samples in our approach is performed as follows:

1. The new sample (protein sequence) is encoded using the selected encoding method.
2. The Euclidean distance between the sample and the centroid of each cluster is found.
3. The cluster that has a minimum distance with the sample is selected and the corresponding SVM is evaluated to predict the new sample.
4. The label of the new sample is determined based on the selected SVM.

7 Experimental Results

7.1 Experiment settings

In this study, two main encoding methods; PseAAC and CTD is used to represent the protein sequences. For PseAAC we set $w = 0.15$ for all experiments and

$\lambda = 30$ [9] for full protein sequences and $\lambda = 3$ for peptide sequences, Two environments are used to implement this work; Java and Matlab, for implementation of k-mean we used the Java machine learning library (Java-ML) [1], the Composition, Transition, Distribution encoding using the Biojava library [23], and we used the Matlab Statistics Toolbox for the SVM implementation [18].

We have constructed a set or preliminary experiments in order to reach the suitable SVM kernels and the corresponding parameters. Among three SVM kernels; linear, polynomial and Radial Basis Function (RBF), we have noticed that RBF led to better results than the two other kernels. The best results were reached when setting γ to 1, and C to 1. It is worth mentioning that any different parameters not considered in the experiments, which may make the SVM classification better than what is reported, is expected to have enhancements on both the existing and the proposed approach since SVM is utilized in both of them.

The specification of our computer that used to run the experiments is as the following: Dell laptop Inspiron 5040, core i5, 8GB RAM.

7.2 Evaluating the performance of the approach

To evaluate the performance of the proposed approach for the new sequences the following steps are performed:

7.2.1 cross-validation step

The main idea of the cross-validation is to split the data, for estimating the risk of each algorithm: part of the data (the training data) is used for training each algorithm, and the remaining part (the testing data) is used for evaluation of the algorithm [2]. The training data in this study is divided into 3 folds.

7.2.2 Clustering step

The training data is grouped into K clusters using a K-mean clustering algorithm, we change K from 2 to 20 clusters and we observed the results.

7.2.3 Distribution step

After the clustering step ends, the samples of each cluster are used for training the SVM. Note that we have hidden some data for testing. For testing we use the following steps:

1. For each cluster, we find a centroid point.
2. Compute the Euclidean distance between each test sample and each centroid of the generated clusters.
3. The testing sample relates to the cluster with the minimum distance between it and the centroids.

Figure 2 illustrates this step. In this figure, the training data is grouped into 3 clusters, each cluster has a centroid, and then the testing data is distributed to the clusters based on the minimum Euclidean distance between the centroids of the clusters and the testing sample.

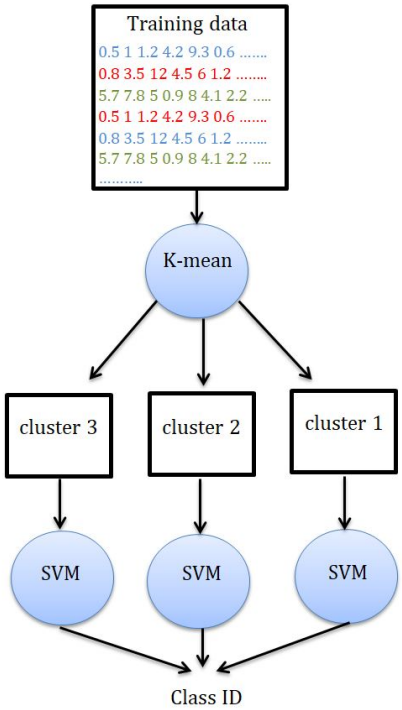


Fig. 2 Distribution step. The distribution of the testing data into three clusters generated from the training data, the distribution is done using the Euclidean distance

7.2.4 Evaluating the performance of the classifier step

There are different methods and evaluating the performance of the classifier. In this study, we used the Area-under-ROC curve (AUC).

A receiver operating characteristics (ROC) curve is a two-dimensional representation of classifier performance, so to compare classifiers we want to reduce ROC performance to a single scalar value representing the performance. A common method is to compute the area under the ROC curve (AUC) [8].

The AUC has an important measuring method, where it is equivalent to the probability that the classifier will rank a randomly chosen positive case higher than a randomly chosen negative case [8].

7.3 Results

To test and demonstrate the proposed algorithm for classification, several experiments were performed. The first experiments were performed using the classification algorithm (SVM) without clustering (K-mean) on Caspase 3 and membrane protein datasets, using the two sets of properties (derived and native).

The second experiments were performed using a clustering algorithm to enhance the classification performance (training time and accuracy), the training

data for each dataset is divided into different numbers of clusters K ranging from 2 to 20.

We then compare the results of the two experiments based on two criteria, the accuracy and time.

7.3.1 classification accuracy results

The first set of experiments was done on Caspase 3 sequences using the two sets of descriptors (derived and native) and three encoding methods (PseAAC, concatenating and CTD methods).

Table 7.3.1 shows the AUC (Area Under Curve) results of applying the proposed approach to the Caspase 3 sequences. The proposed approach led to an improvement in classification accuracy for Caspase 3 dataset when the encoding methods were the concatenating method and PseAAC. The last column in the table represents the number of clusters that achieved the higher AUC in the case of using clustering before the classification.

The AUC values of Caspase 3 sequence. The AUC values for using two encoding methods and two sets of properties with and without clustering before the classification.

Encoding method	Descriptors	AUC using classification without clustering	Classification using clustering	
			AUC	Number of clusters (K)
Concatenating method	Derived properties	0.56	0.85	4
	Native properties	0.56	0.86	7
PseAAC method	Derived properties	0.58	0.63	5
	Native properties	0.55	0.64	19
CTD method		0.59	0.57	20

The results of applying the proposed approach using the concatenating method in Table 7.3.1 show a 34% improvement on AUC over the classification without clustering while using the PseAAC show a 12% improvement.

In Table 7.3.1 we present the best results obtained in term of AUC for the number of clusters using the proposed approach. Figure 3 illustrates the ROC Curves resulted from applying the concatenating method with native properties with and without using the clustering algorithm.

The results show that using the CTD for the proposed approach did not improve the AUC values for Caspase 3 dataset.

The second set of experiments was done on membrane protein sequences using the two sets of descriptors and two encoding methods (PseAAC, and CTD methods).

Table 7.3.1 shows the AUC results of applying the proposed approach to the membrane proteins. The proposed approach led to an improvement in classification accuracy when using the two encoding methods. Figure 4 illustrates the ROC Curves resulted from applying PseAAC method with derived properties with and without clustering.

The results of applying the proposed approach using PseAAC method in Table 7.3.1 shows about 6% improvement on AUC over the classification without clustering, while using the CTD method shows an 11% improvement.

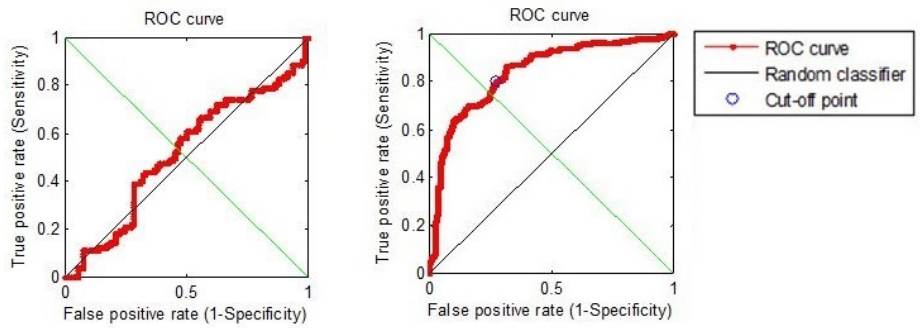


Fig. 3 Roc Curves resulted from applying concatenating method with native properties on Caspase 3 sequences. Left, represents the ROC curve resulted from applying classification without applying clustering. Right, represents the ROC curve resulted from applying classification with clustering.

The AUC values of membrane proteins sequence. The AUC values for using two encoding methods and two sets of properties with and without clustering before the classification.

Encoding method	Descriptors	AUC using classification without clustering	Classification using clustering	
			AUC	Number of clusters (K)
PseAAC method	Derived properties	0.76	0.84	20
	Native properties	0.79	0.84	18
CTD method		0.51	0.62	3

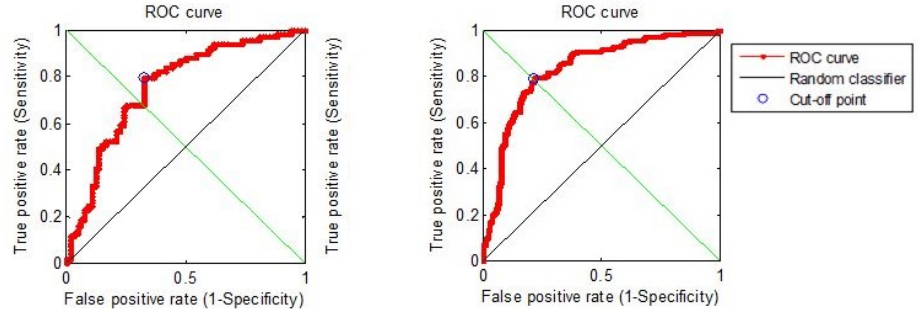


Fig. 4 ROC Curves resulted from applying PseAAC method with derived's properties on membrane protein sequences. Left, represents the ROC curve resulted from applying classification without clustering. Right represents the ROC curve resulted from applying classification with clustering.

7.3.2 Time performance

Many previous studies have focused on reducing the computation time for large datasets by using the clustering before classification approach because training an SVM is usually posed as a quadratic programming (QP) problem to find a hyperplane which implicates a matrix of density $n \times n$, where the n is the number

of samples in the dataset, so the training complexity of SVM is dependent on the size of a dataset [6].

Instead of having a time complexity of $O(n^2)$, we can reduce it by breaking down the whole training set into a set of clusters. If we assume the clusters are nearly equal and each cluster has a data set of size m, where $m < n$. Then the time complexity will become $O(c \cdot m^2)$, where c is the number of clusters.

In order to study the time performance of our approach, the approach was applied to two datasets with a different number of clusters (range from 2 to 10 clusters). The native properties were used by the PseAAC encoding method.

Figure 5 illustrates the change of the time (in second) with the increase in the number of clusters for the Caspase 3 and membrane protein datasets. The figure shows that the time decreased from 0.24 to 0.05 seconds, the time continued to decline at three clusters, then it began to increase slightly, this increase is due to the overhead caused by increasing the number of clusters.

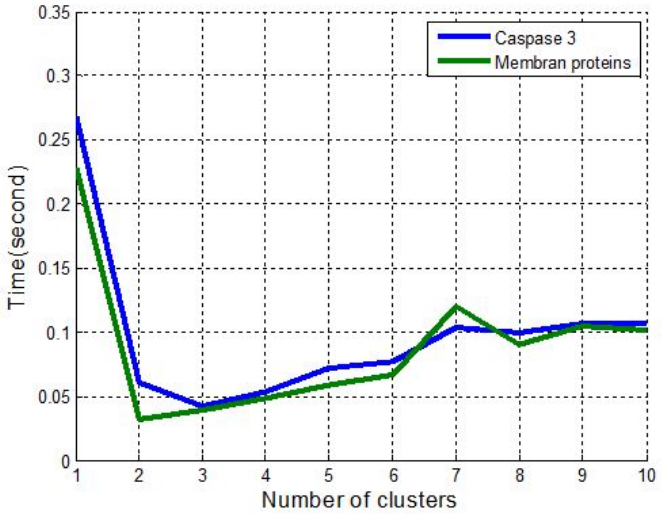


Fig. 5 The time performance for the proposed approach based on SVM algorithm. The experiment was done using the Caspase 3 and membrane protein benchmarks, the PseAAC encoding method and the Native properties were used

8 Discussion

Based on previous results, the proposed approach that depended on applying the clustering before the classification has proven to enhance the accuracy of classification for the two benchmarks; Caspase 3 and membrane proteins.

In this study, we used K-mean for clustering and SVM for classification for all experiments. When applying PseAAC to classification without clustering, it was clear that the classification rates were not satisfactory (around 53%) for Caspase 3 sequences, but it gave good results for the membrane proteins (around 77%). On

the other hand, the accuracy for the membrane proteins and Caspase 3 sequences were improved significantly when the clustering was introduced before the classification (around 63% for Caspase 3 and around 84% for membrane proteins). The accuracy of sequences for the two sets of properties increased when the data were divided into two clusters. After that, the values of accuracy swing up and down but it remained higher than the accuracy of the classification without clustering.

The CTD of classification without clustering failed to give good results neither for the membrane proteins nor for Caspase 3 sequences. When the clustering was used before the classification, the results of CTD were not enhanced in the case of Caspase 3, but it made improvement of 11% in the case of membrane proteins.

The concatenation method was only applied to Caspase 3 sequences because they have fixed lengths, the results were not good in the case of classification without clustering. When using clustering before classification the concatenation method gave better results than PseAAC and CTD, that is because it uses the natural values of the PCPs and that made the differences between the selected sets of properties clear, while almost sets of properties behaved the same based on PseAAC, that's because the PseAAC depends on the features that derived from the natural PCPs, so the values will be close for the same dataset. Figure 6 illustrate how the AUC values changed when using the concatenation method for Caspase 3 sequences using a different number of clusters.

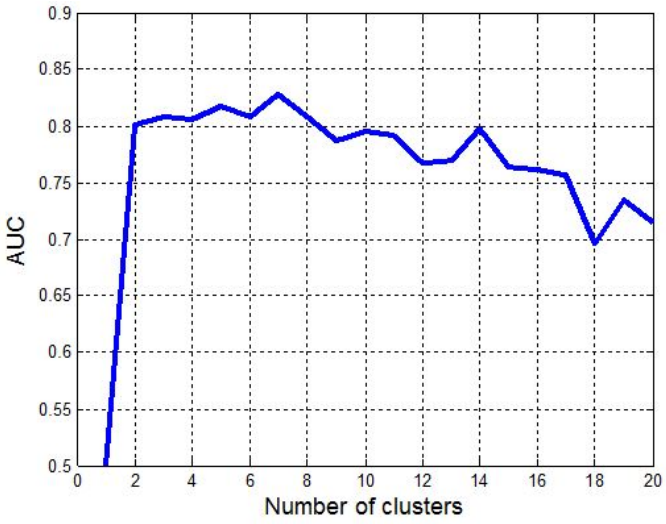


Fig. 6 AUC values of SVM for Caspase 3 sequences using the concatenating method with native properties. The training data divided into different numbers of clusters (range from 2 to 20), one cluster of training data means a classification without clustering

In most of the above experiments, we used 2 sets of descriptors; native properties and derived properties that were proposed by Georgiev [10]. This study shows that the AUC values for using the native properties and derived properties gave the same results for all experiments, this means the descriptors of Georgiev are excellent representatives for the 50 native properties.

The result of the time performance on the two datasets showed that our approach significantly reduced the training time of the SVM while improving the accuracy of the prediction, and without eliminating any samples.

9 Conclusion

This study has evaluated a recent method for enhancing the accuracy of the classification of the heterogeneous protein data. This approach is based on the using of clustering algorithm before the classification, using two sets of descriptors based on PCPs, and applying two encoding methods to represent the sequences. The results show that the classification based on the clustering can be significantly enhanced the accuracy of the prediction for the protein sequences, and this enhancement depends on the selected PCPs and the encoding methods used to represent the sequences, this mean that the datasets of the proteins need to examine again to distribute the sequences based on their similarities, in order to facilitate the classification.

Also, the proposed approach succeeds in reducing the training time of the SVM significantly while improving the accuracy of prediction. That means our approach can be used to reduce the SVM training time for large datasets, without the need to eliminate any sample from the dataset as in previous approaches. This result is consistent with the previous studies such as Cervantes, et al. [6].

10 Future Works

In the future, other encoding methods and other descriptors can be used to enhance the results of our approach, also different clustering and classification techniques can be used rather than the K-mean and SVM.

The most important outcome of our approach is to develop a tool depending on this approach in order to help the researcher to know which descriptors, encoding method, clustering and classification algorithms can be used to enhance the accuracy of the prediction for different datasets of proteins.

Based on this approach the researchers in the future can experimentally determine the best descriptors for each dataset (that achieve the higher accuracy).

References

1. T. Abeel, Y. Peer, and Y. Saeys. Java-ml: A machine learning library. *Journal of Machine Learning Research*, 10:931–934, 2009.
2. S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *statistics Surveys*, 4:40–79, 2010.
3. M. Awad, L. Khan, F. Bastani, and I. Yen. An effective support vector machines (svm) performance using hierarchical clustering. *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, pages 663–667, 2004.
4. M. Ayyash, H. Tamimi, and Y. Ashhab. Developing a powerful in silico tool for the discovery of novel caspase-3 substrates: a preliminary screening of the human proteome. *BMC Bioinformatics*, 2012.
5. András Bánhalmi, Róbert Busa-Fekete, and Balázs Kégl. A one-class classification approach for protein sequences and structures. In *International Symposium on Bioinformatics Research and Applications*, pages 310–322. Springer, 2009.

6. J. Cervantes, X. Li, and W. Yu. Support vector machine classification based on fuzzy clustering for large data sets. *MICAI'06 Proceedings of the 5th Mexican international conference on Artificial Intelligence*, pages 572–582, 2006.
7. C. Chou. Prediction of protein cellular attributes using pseudo-amino-acid composition. *PROTEINS: Structure, Function, and Genetic*, pages 246–255, 2001.
8. T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27:861–874, 2006.
9. Q. Gao, Ye X., Jin Z., and J. He. Improving discrimination of outer membrane proteins by fusing different forms of pseudo amino acid composition. *Analytical Biochemistry*, 398:52–59, 2010.
10. A. Georgiev. Interpretable numerical descriptors of amino acid space. *JOURNAL OF COMPUTATIONAL BIOLOGY*, 16(5), 2009.
11. S. Gunn. Support vector machines for classification and regression. *Technical Report*, 1998.
12. S. Hellberg, M. Sjostrom, and S. Wold. The prediction of bradykinin potentiating potency of pentapeptides. an example of a peptide quantitative structure–activity relationship. *Acta Chem. Scand.*, pages 135–140, 1986.
13. Pooja Jain and Jonathan D Hirst. Automatic structure classification of small proteins using random forest. In *BMCBI*, 2010.
14. S. Kawashima and M. Kanehisa. Aaindex: Amino acid index database. *Nucleic Acids Research*, 27:27–36, 1999.
15. A. Kyriakopoulou and T. Kalamboukis. Text classification using clustering. *In Proceedings of the ECML-PKDD Discovery Challenge Workshop*, 2006.
16. Roman A. Laskowski, Janet M. Thornton, and Michael J.E. Sternberg. The fine details of evolution. *Biochemical Society Transactions*, 37(4):723–726, 2009.
17. V. Mathura and D. Kolippakkam. Apdbase: Amino acid physicochemical properties database. *Bioinformation*, 1, 2005.
18. The Mathworks. Statistical toolbox 7.0. http://www.mathworks.com/help/stats/index.html.
19. M. McKee and J. McKee. *Biochemistry: The Molecular Basis of Life*. Oxford University Press, USA, 5 edition, 2011.
20. L. Nanni, S. Brahnam, and A. Lumini. High performance set of pseaac and sequence based descriptors for protein classification. *Journal of Theoretical Biology*, 2010.
21. S. Ong, H. Lin, Y. Chen, Z. Li, and Z. Cao. Efficacy of different protein descriptors in predicting protein functional families. *BMC Bioinformatics*, 2007.
22. K. Park, Gromiha M., P. Horton, and M. Suwa. Discrimination of outer membrane proteins using support vector machines. *Bioinformatics*, 21:223–229, 2005.
23. A. Prlic, A. Yates, and et al. Bliven, S. Biojava: an open-source framework for bioinformatics. *Bioinformatics*, 28:2693–2695, 2012.
24. Rojas R. Neural networks: A systematic introduction. *Springer-Verlag, Berlin*, 1996.
25. A. Rahideh and M. Shaheed. Cancer classification using clustering based gene selection and artificial neural networks. *2nd International Conference on Control, Instrumentation and Automation (ICCIA)*, 2011.
26. S. Ray and T. Kepler. Amino acid biophysical properties in the statistical prediction of peptide-mhc class i binding. *Immunome Research*, 2007.
27. R. Saidi, M. Maddouri, and E. Nguifo. Protein sequences classification by means of feature extraction with substitution matrices. *BMC Bioinformatics*, 2010.
28. P. Sneath. Relations between chemical structure and biological activity in peptides. *Journal of Theoretical Biology*, 12:157– 195, 1996.
29. Yan Yuan Tseng and Wen-Hsiung Li. Classification of protein functional surfaces using structural characteristics. *Proceedings of the National Academy of Sciences*, 109(4):1170–1175, 2012.
30. Y. Xiong, J. Liu, W. Zhang, and T. Zeng. Prediction of heme binding residues from protein sequences with integrative sequence profiles. *Proteome Science*, 10, 2012.
31. H. Yu, J. Yang, and J. Han. Classifying large data sets using svms with hierarchical clusters. *ACM, Knowledge Discovery and Data Mining conference*, 2003.