Improved Protein Function Prediction by Combining Clustering with Ensemble Classification

Haneen Altartouri and Tobias Glasmachers
Ruhr-University Bochum, Germany
Email: {haneen.altartouri, tobias.glasmachers}@ini.rub.de

***Abstract*—Predicting protein functions is a challenging task in bioinformatics, different machine learning algorithms have been used for this task. In this paper, we investigate the effect of applying clustering and ensembles of classifiers to improve the performance of the prediction. Two approaches are proposed, the first approach depends on clustering to build an ensemble of classifiers, while the second approach uses the clustering to break down the complex dataset into sub-datasets, then an ensemble of different classifiers train inside each sub-dataset. We observed that this combination of clustering and classifications improved the performance of prediction in the most cases.**

***Index Terms*—protein function classification, clustering, stacking, diverse classifiers**

I. Introduction

Technological advances have increased the rate at with new protein sequences are discovered. As a consequence, it has become necessary to identify protein functions accurately in a cost-effective manner. However, determining the functions experimentally is expensive and time consuming, so researchers depended on computational approach to predict the function of proteins from the sequences [1].

Most of the computational methods for protein prediction are based on classification algorithms. However, in some cases it is difficult to achieve the desired performance using classification alone. Some approaches attempt to improve the performance of classification by combining one or more different classifier algorithms into an ensemble of classifiers [2], [3]. The diversity of classifiers can be enriched by varying the parameters of the classifier [4]. These methods lead to Ensemble learning. Many studies proved that ensembles of diverse classifiers improve classification accuracy of a single classifier [5]-[7].

The most popular methods to ensemble predictions from different models are bagging, boosting and stacking. With bagging, multiple models are built from different sub-samples of the training set [8]. Boosting generates a sequence of models, each of which learns to fix the prediction errors of the previous models in the chain [9]. Stacking combines the prediction results of base models, which can be different types of models, using a meta-classifier (supervisor model), which learns how to best combine the predictions of the base models [10].

Another way to improve performance is to combine structural information extracted from clustering with the classification in different ways. Previous works showed that using clustering with classification can improve the performance in different applications; such as disease diagnosis [11], network traffic classification [12], and activity recognition within smart environments [13]. One motivation for combining clustering with classification is to reduce the heterogeneity of the dataset by breaking down a complex classification problem into simpler problems using clustering, then training a single classifier on each cluster [14], [15]. Other approaches focused primarily on using clustering to increase the number of classes and the classifier trained to distinguish between these new classes [16].

Since both ensembles and clustering have the potential to improve classification performance of protein sequence prediction, combining the two techniques is a promising approach. In this work we systematically investigate two ways to combine ensemble learning and clustering.

Combinations of ensemble learning and clustering are not entirely new. Some researches improved the classification by combining an ensemble of classifiers (one or more classifiers) with the information from an ensemble of clusters (one or more clusters) to get new properties for improving the prediction of new data [17]-[19]. Acharya *et al.* combined ensembles of classifiers and ensembles of clusters to generate a consolidated classification. In their work, an ensemble of classifiers is first learned on the labeled training dataset to get initial class probability distributions, these probabilities represent unlabeled data for the next step. Then, a cluster ensemble is applied to the probabilities data to get a similarity matrix that is used to update the initial class probability distributions obtained from the classifier ensemble [17]. Others applied clustering on the training set to generate a set of diverse classifiers. Trivedi *et al.* applied k-means to group the training data into clusters. They varied the number of clusters to get different sets of

Manuscript received December 1, 2020; revised May 22, 2021.

doi: 10.12720/jait.12.3.197-205

clusters. Linear regression was used inside clusters and averaged the results. They applied this approach on different regression problems, and the results showed a significant improvement in datasets with a cluster structure [20].

As far as we know, the combination of ensemble learning and clustering to improve the performance of prediction for protein functions has not yet been studied. This research uses clustering and ensembles of classifiers to improve the performance of the prediction for protein functions. We investigate two approaches to improve the performance:

1) The first approach uses stacking globally, at the level of the full dataset. It depends on clustering to build an ensemble of diverse classifiers. The outputs of these classifiers are fed into the meta-classifier to find the best combination for the final prediction.
2) The second approach uses stacking locally, at the level of individual clusters. We use clustering to break down the complex dataset into homogeneous groups (sub-datasets), and handle each sub-dataset as a small problem inside the complex dataset. We train an ensemble of diverse classifiers inside each sub-dataset to get a more powerful and robust model. To help weak clusters, we always include a model trained on the whole dataset in the ensemble.

We evaluate proposed approaches on six protein function prediction problems. Our results show that both proposed approaches improve the performance of prediction in the most cases.

The reminder of the paper is organized as follows: Section II describes the proposed approaches in detail. Section III briefly introduces the benchmarks of this study. In Section IV, we present experimental results and discuss the findings. The last is the conclusions of the work.

II. The Proposed Approaches

We propose two approaches to improve the performance of classifying protein functions based on combinations of ensemble learning and clustering. In all cases, our processing chain involves the following steps:

- We first need to represent protein sequences in a form that can be easily handled by classification and clustering algorithms. This means in particular that variable length sequences need to be represented by fixed-length feature vectors. In this study we use Chou's Pseudo Amino Acid Composition (PseAAC) descriptor [21], which is widely used in previous researches [22], [23]. A sequence is represented by $20 + \lambda$ numerical features. The first 20 features are the frequencies of the 20 amino acids. The remaining λ descriptors represent the sequence order. For a detailed description of PseAAC please refer to [21], [24]. PseAAC depends on Physico-Chemical Properties (PCPs) of the amino acids. In this work we use two standard sets of PCPs. The first set consists of three PCPs used in Chou's work [21]: hydrophobicity, hydrophilicity, and side chain mass. The other set contains fifty non-redundant PCPs of amino acids proposed by Georgiev [25].
- We always train a classifier on the whole dataset, called Full Dataset Classifier (FDC). In this study we train different classifiers on the protein sequences represented by PseAAC descriptors using 50 PCPs and 3 PCPs. The experiments showed that Support vector Machine (SVM) [26] is the best choice in most datasets using 50 PCPs, while Random Forest (RF) [27] is the best when using 3 PCPs. Therefore, in this research we use SVM and RF classifiers on the corresponding feature sets.
- We cluster the dataset into groups (sub-datasets) with the k-means algorithm. K-means is a simple and easy-to-implement algorithm. It is widely used in bioinformatics researches [28], [29]. For more details on k-means please refer to [30]. We vary the number of clusters (k) to obtain diverse subsets and subclassifiers. In our experience, the use of alternative clustering algorithms does not change the results significantly.

In the following two sections we describe our approaches to combining ensembles (stacking) and clustering in detail.

A. Combining Cluster Predictions Approach

This approach uses clustering to obtain a diverse set of FDC classifiers, which are then stacked. Fig. 1 illustrates the proposed approach.

For each value of k, the dataset is split into k clusters or subdataset. A separate classifier is trained on each sub-dataset, including tuning of its hyperparameters. The k local classifiers are combined into a FDC by directing each test point to its cluster and predicting its label with the corresponding local classifier. Each local classifier delivers class probabilities for further processing. We used SVM and RF models for the sub-datasets.

In order to obtain diverse predictions, we vary the parameter k of k-means from 1 to some n, where $k = 1$ represents the whole dataset. We collect the prediction probabilities generated from different values of k and treat them as new features.

These predictions (features) can be combined in different ways to get the final prediction, such as (weighted) averaging [20]. In this study we handle these new features as a new data set. The final predictions are generated by training a new classifier on this set, called meta-classifier. This is similar to the second layer in stacking approaches, where the meta-classifier is trained to ideally combine the model predictions to get the final predictions [10].

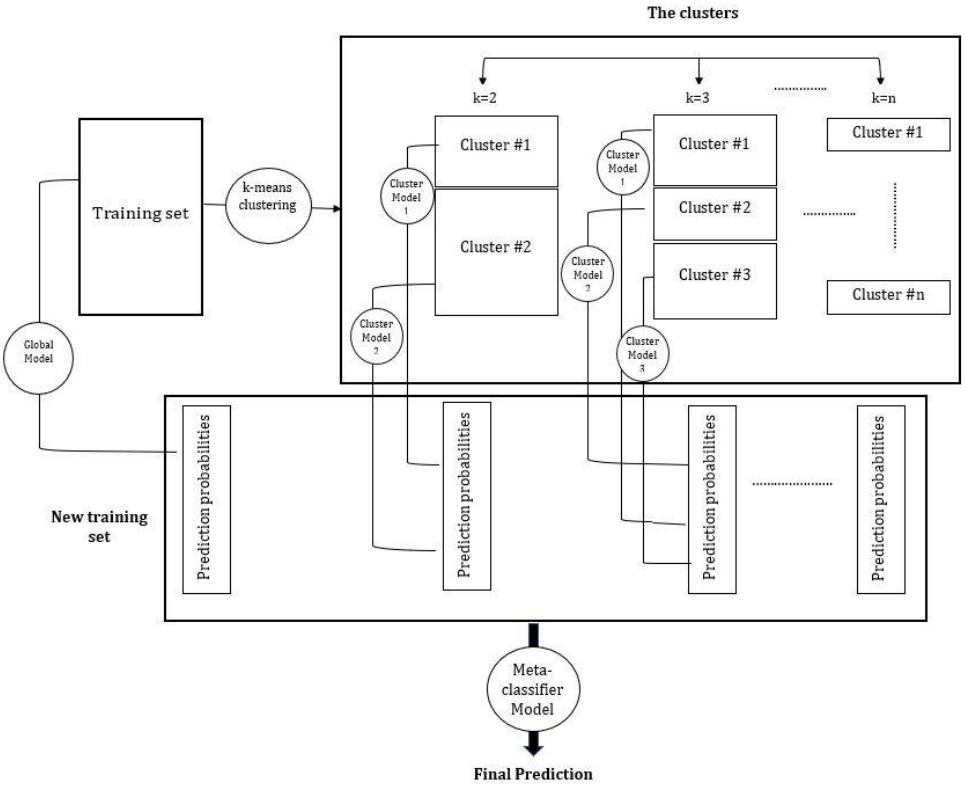

Figure 1. Combining cluster predictions approach.

Different classifier types were studied as a meta-classifier, and we found the logistic regression [31] outperformed the other classifiers in most cases. That is because there is almost a linear relationship between the probabilistic outputs in the new training set. We therefore restrict ourselves to logistic regression in the following.

B. Inter-Cluster Stacking Approach

Our second approach relies on breaking down a complex classification problem into simpler and more homogeneous problems using clustering. We then employ stacking inside each cluster. The processing logic is illustrated in Fig. 2.

After applying K-Means, we apply stacking inside each cluster by using an ensemble of classifiers instead of a single classifier. The ensemble shall improve and stabilize the performance of prediction inside each cluster. The success of stacking relies on applying diverse models in the clusters. Therefore, we train an ensemble of linear and non-linear models as base classifiers for stacking inside the clusters: SVM, RF, Artificial Neural Network (ANN), eXtreme Gradient Boosting (xGBoost), logistic regression, and K-Nearest Neighbors (KNN). For a detailed description of these algorithms we refer to [26], [27], [31]-[34], respectively. During the training we take into consideration tuning the hyper-parameter for these classifiers.

The predictions generated from the base classifiers are used as inputs to the meta-classifiers in each cluster. Different classifiers are tested as a meta-classifier, and as in the first approach, logistic regression was found to be superior to other classifiers in the most cases.

In some cases, the performance of the individual classifiers is not good based on specific criteria (such as accuracy) compared to the performance of the FDC on the same region. In this case, Fradkin proposed an option to return back to the FDC [16]. However, there is no optimal criterion that can be used to determine when to use FDC or cluster classifier for a specific region. So, to avoid the decision problem we combined the probabilistic outputs generated by FDC for the samples in each cluster with the outputs of base classifiers trained on the same cluster, as shown in Fig. 2, and leave the decision to the stacking layer. This soft combination approach also bears the potential to outperform a hard decision. Furthermore, some clusters suffer from a lack of sufficient data to train classifiers locally, resulting in severe overfitting. In such cases we use FDC only to get the predictions for the samples belong to these small clusters.

III. BENCHMARK DATASETS

To evaluate the performance of the proposed approaches, we have used six datasets, three of which are peptide sequences: Caspase 3 human substrates, Antimicrobial peptides (AMP), and Major Histocompatability Complex II (MHCII), the other three datasets are long sequences: DNA-binding proteins, Antioxidant proteins, and RNA-binding proteins. Table I summarizes the datasets. For more details please refer to [35]-[40], respectively. We have split each dataset into a training and a testing set.

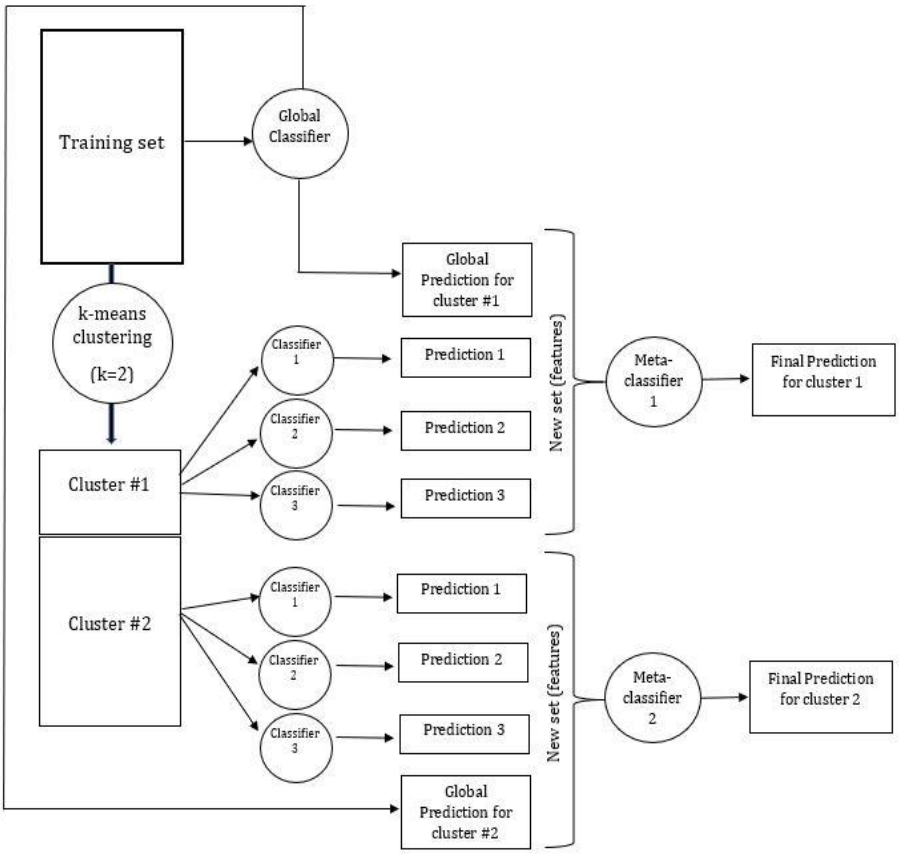

Figure 2. Inter-Cluster stacking approach.

TABLE I. DATASETS USED FOR THE APPROACH EVALUATION

Dataset	# of Positives	# of Negatives
DNA-binding proteins	523 binding proteins	543 non-binding proteins
Antioxidant proteins	250 antioxidants	1547 non-antioxidant
RNA-binding proteins	2780 binding proteins	7077 non-binding proteins
AMP	869 AMPs	2405 non-AMPs
Caspase 3 human substrates	247 cleaved peptides	247 non-cleaved peptides
MHCII	3510 binding peptides	1656 non-binding peptides

IV. EXPERIMENTS AND RESULTS

To evaluate the efficacy of the proposed approaches, we compared the performance of these approaches with the FDC (either SVM based on 50 PCPs or RF based on 3 PCPs), which is a natural baseline.

In our experiments, sequences were encoded using PseAAC method as described in Section II using the following parameters: The weight of the features representing sequence order was set to $w = 1/2$. The length of the shortest sequence was set to $\lambda = 7$ for peptides and to $\lambda = 30$ for long protein sequences. These settings result in 27 and 50 features, respectively.

The number of clusters k was varies as follows. For small datasets (Caspase-3, DNA-binding, and Antioxidant proteins) k is varied in the range from 2 to 7 with a step size of 1, and for the other datasets k is varied in the range from 5 to 30 with a step size of 5.

We have tuned the hyper-parameters of all classifiers (FDC, base classifiers, meta-classifiers) using 5-fold cross-validation repeated 3 times.

To evaluate the performance of our approaches we rely on sensitivity (SN), specificity (SP), and Matthew's Correlation Coefficient (MCC) [41]. In addition, we considered the area under the Receiver Operating Characteristics (ROC) curve, or area under the ROC curve (AUC) for short [42]. All reported values are computed on independent test sets.

We present different types of results. Fig. 3, 4, 5, and 6 focus on the effect of varying the number of clusters k for the proposed approaches, for SVM classifiers on the sequences encoded using 50 PCPs. Table II compares our two approaches to the baseline. It shows the best results we have achieved for both using SVM and RF, as well as the results corresponding to the hyperparameters that yield the best validation errors, and which are hence be selected by a standard model selection procedure.

TABLE II. COMPARISON BETWEEN APPLYING FDC ONLY, COMBINING PREDICTIONS OBTAINED BY VARYING *k* VALUE, AND STACKING INSIDES CLUSTERS FOR 6 BENCHMARKS. THE RESULTS REPRESENT THE BEST VALUE OF *k* BASED ON THE VALIDATION (VALID.) AND THE CORRESPONDING TEST RESULTS (TEST)

Method	SVM						RF					
	k	**Set**	**AUC**	**SEN**	**SPE**	**MCC**	***k***	**Set**	**AUC**	**SEN**	**SPE**	**MCC**
DNA-binding proteins												
- FDC only (the baseline)	-	-	0.8033	0.7769	0.6963	0.4744	-	-	0.7899	0.6692	0.7556	0.4266
- Combining cluster predictions	1,2,3,4	(Valid.)	0.8307	0.7634	0.7525	0.5157	1,2	(Valid.)	0.8003	0.8203	0.7841	0.5037
		(Test)	0.8256	0.7538	0.7481	0.5019		(Test)	0.7978	0.6769	0.8074	0.489
- Inter-cluster stacking	4	(Valid.)	0.8406	0.7888	0.75	0.5389	6	(Valid.)	0.8058	0.6743	0.8333	0.5149
		(Test)	0.8252	0.7769	0.7481	0.5251		(Test)	0.8025	0.6692	0.8148	0.4899
Antioxidant proteins												
- FDC only (the baseline)	-	-	0.8405	0.68	0.8987	0.5193	-	-	0.8493	0.7419	0.8627	0.5032
- Combining cluster predictions	-	-	no improvement achieved				1,2,3	(Valid.)	0.8692	0.7872	0.8765	0.5587
		-						(Test)	0.8528	0.7581	0.8679	0.5228
- Inter-cluster stacking	2	(Valid.)	0.8565	0.8085	0.8398	0.5181	7	(Valid.)	0.9006	0.7713	0.9208	0.6298
		(Test)	0.8547	0.7903	0.8161	0.4728		(Test)	0.8877	0.7581	0.9197	0.6172
RNA-binding proteins												
- FDC only (the baseline)	-	-	0.903	0.6331	0.9582	0.6548	-	-	0.9053	0.636	0.9661	0.6727
- Combining cluster predictions	1,5,10	(Valid.)	0.9348	0.742	0.9432	0.712	1,5	(Valid.)	0.9088	0.6803	0.9655	0.7053
		(Test)	0.9286	0.7252	0.9435	0.6995		(Test)	0.9075	0.6691	0.957	0.6806
- Inter-cluster stacking	10	(Valid.)	0.9284	0.7463	0.9495	0.7263	5	(Valid.)	0.9378	0.7254	0.9642	0.7371
		(Test)	0.9278	0.7324	0.948	0.7129		(Test)	0.9339	0.7151	0.9559	0.7139
AMP peptides												
- FDC only (the baseline)	-	-	0.9552	0.765	0.9418	0.7247	-	-	0.9624	0.7926	0.9484	0.7574
- Combining cluster predictions	1,5	(Valid.)	0.9936	0.9064	0.9119	0.7846	1,5,10	(Valid.)	1	0.8466	0.9483	0.7975
		(Test)	0.9729	0.8986	0.9068	0.771		(Test)	0.9867	0.8341	0.9384	0.7714
- Inter-cluster stacking	5	(Valid.)	0.9638	0.7055	0.9202	0.6416	15	(Valid.)	0.9668	0.8333	0.9583	0.791
		(Test)	0.9638	0.6959	0.9185	0.631		(Test)	0.9648	0.8111	0.9517	0.7771
Caspase 3 peptides												
- FDC only (the baseline)	-	-	0.7487	0.623	0.7541	0.3803	-	-	0.7263	0.7377	0.5246	0.2685
- Combining cluster predictions	1,2	(Valid.)	0.7756	0.7276	0.7276	0.4552	1,2,3	(Valid.)	0.7344	0.6057	0.767	0.3777
		(Test)	0.7617	0.6885	0.7541	0.4436		(Test)	0.7329	0.7541	0.5902	0.349
- Inter-cluster stacking	3	(Valid.)	0.7701	0.7273	0.6935	0.4208	7	(Valid.)	0.7727	0.7727	0.6452	0.4206
		(Test)	0.7525	0.6885	0.7213	0.4101		(Test)	0.7603	0.7705	0.6393	0.4134
MHCII peptides												
- FDC only (the baseline)	-	-	0.8034	0.7605	0.6981	0.4396	-	-	0.7909	0.7571	0.7029	0.4401
- Combining cluster predictions	1,5	(Valid.)	0.8244	0.8297	0.6656	0.492	-	(Valid.)	no improvement achieved			
		(Test)	0.8241	0.8119	0.6739	0.4773		(Test)				
- Inter-cluster stacking	20	(Valid.)	0.8152	0.7976	0.7552	0.5316	20	(Valid.)	0.811	0.771	0.7287	0.4782
		(Test)	0.8019	0.7834	0.7464	0.5077		(Test)	0.7919	0.764	0.7222	0.4648

A. Impact of Combining Predictions Obtained by Varying (k) Value

We ran two sets of experiments to study the effect of the approach proposed in Section II.A to improve the performance of protein predictions, using 1) SVM classifiers and 2) RF classifiers. The results show clearly that the proposed approach improves upon the FDC performance, especially when the sequence is encoded using 50 PCPs and SVMs are trained within the clusters.

Fig. 3 and 4 show a comparison between using FDC only (baseline, $k = 1$), and combining of different cluster predictions, where SVMs are used. Fig. 3 shows AUC values of the models, and Fig. 4 shows the corresponding

MCC values. The vertical axis represents the models combined to generate a single model. We displayed the results of combining up to k = 5 models for small datasets and up to k = 20 models for the other datasets, since there is no improvement achieved when combining more models.

We observed that in most cases combining diverse models improves the performance over using FDC only, except for the Antioxidant proteins, where adding models to the FDC reduced the performance when SVM is used inside the clusters (see Fig. 3 and 4). Training RF inside the clusters, and combining FDC with models generated from k = 2 and k = 3 improved the performance of Antioxidant proteins very slightly (1% improvement in the AUC value and 2% improvement in MCC, see Table II).

Applying the proposed approach improved the AUC by about 2% for all datasets except Antioxidant, while for MCC we improved upon the baseline by about 3% to 6%. The best results for peptide datasets (Caspase 3, AMP, and MHCII) are achieved by combining predictions of two models. One of them is the FDC model, while for long protein sequences (RNA-binding, and DNA-binding) it is required to combine three or four models to achieve the best results. Adding more models does not improve performance further, but rather reduces the performance of the combined model. This is because at high value of k we did not get a good structure for some clusters due to a lack of data, which leads to adding noise to the combined model.

Table II shows that applying RF inside the clusters improves the performance for most datasets except for MHCII peptides. For all datasets the improvement in the AUC is small (about 1%). However, we have achieved a significant improvement for the other metrics. For the DNA-binding dataset we achieved an improvement of 6% for MCC by combining the predictions of FDC with models generated from k = 2. For Caspase 3 and Antioxidant we have achieved the best results (8%, and 2% for MCC respectively) by combining FDC with models generated from k = 2 and k = 3. For RNA-binding and AMP the improvement is about 2% in MCC.

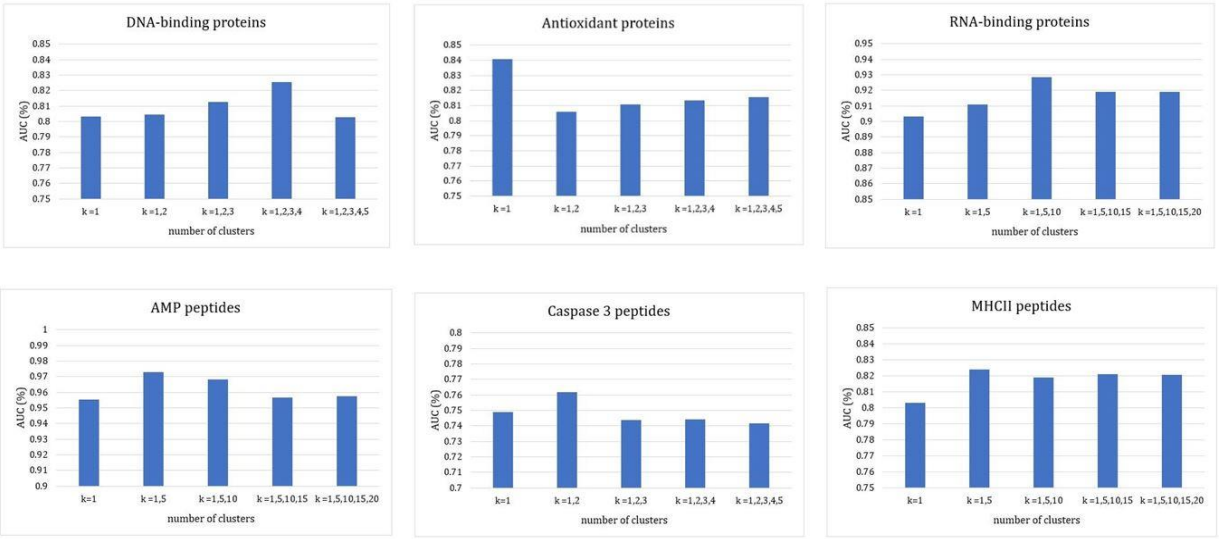

Figure 3. Comparison between AUC values for the FDC and combining predictions obtained by varying (k) value.

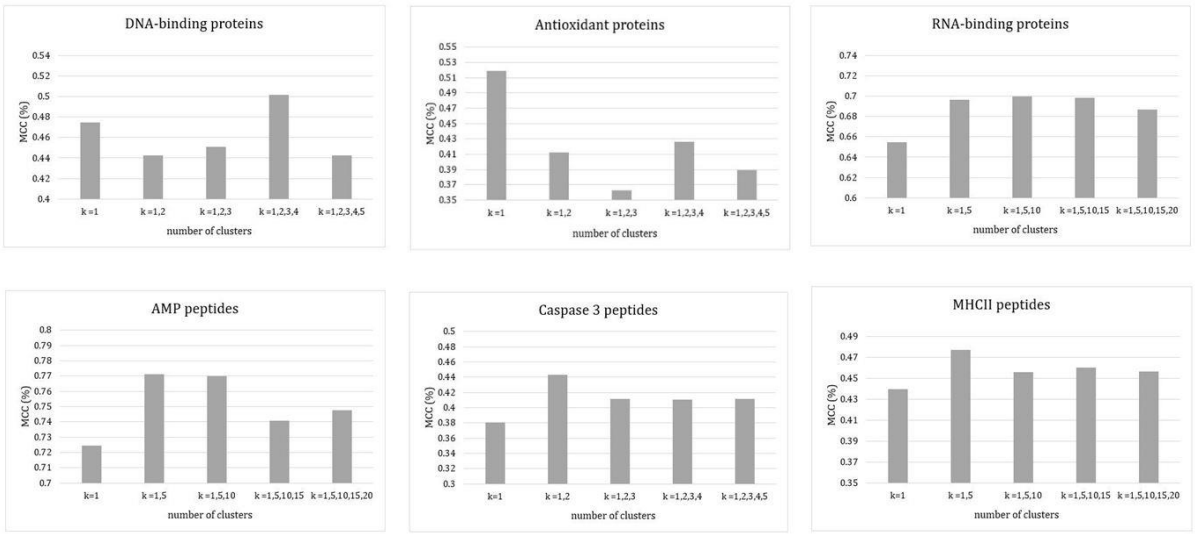

Figure 4. Comparison between MCC values for the FDC and combining predictions obtained by Varying (k) value.

B. Impact of Applying Stacking within the Clusters

We ran similar experiments to study the effect of the approach proposed in Section II.B. The results indicate that the proposed approach significantly improved the performance over the FDC especially when RF is used, see Table II.

Fig. 5 and 6 show a comparison between using FDC only (baseline, k = 1), and clustering with stacking inside the clusters at different values of k, where SVM is used for FDC. These figures indicate that we could not improve the performance at high value of k, where some clusters suffer from poor structure. The improvement of AUC for all datasets is about 1% - 2%, while we have achieved a significant improvement in MCC except for Antioxidant and AMP peptides, for which applying the proposed approach reduced the MCC compared to FDC only.

For DNA-binding we obtained the best result by grouping the dataset into 4 clusters (2% for AUC, and 5% for MCC). For RNA binding proteins we have achieved 2% for AUC and 6% for MCC. On the other hand, for MHCII peptides, while the approach did not improve the overall AUC of the classifier, is yields a good improvement in the MCC value (about 7%).

Table II shows that using RF, the baseline results significantly improved in both AUC and MCC. For the DNA-binding dataset we achieved an improvement of 2% for AUC and 6% for MCC by splitting the dataset into 6 clusters and applying stacking inside each cluster. For Antioxidant we have achieved 4% improvement for AUC and 11% for MCC. For RNA-binding we have achieved 3% and 4% improvement for AUC and MCC by grouping the dataset into 5 clusters.

For AMP and MHCII, we did not improve the overall AUC of the classifier, while we achieved 2% improvement in MCC. The best result is obtained for Caspase: 4% and 15% improvement for the AUC and MCC values, by grouping the dataset into 7 clusters.

V. Discussion

In the most cases, applying the proposed approaches improves the performance over baseline. This is achieved by the combination of extracting structurally meaningful cluster information with the power of ensemble learning.

Both approaches showed that for small values of k we can gain useful information about the structure of the data, which improve the prediction performance. For high value of k, especially for small dataset, clusters tend to be small, so classifiers are easily prone to overfitting. This finding supports the common sense to use only few small values for k (and hence for n).

We have found that in both of our approaches a linear model is sufficient for combining class probabilities in the meta classifier stage. This makes logistic regression applicable, which is beneficial since it does not require hyperparameter tuning. In any case, we believe that applying a meta-classifier is generally superior to heuristic decision rules and to manual tuning of weights when combining the outputs of multiple classifiers.

The encoding method of representing the protein sequences naturally affects the performance of the proposed approaches. This is because the performance of this kind of approaches depends on the clusterability of the protein datasets depends on the encoding method.

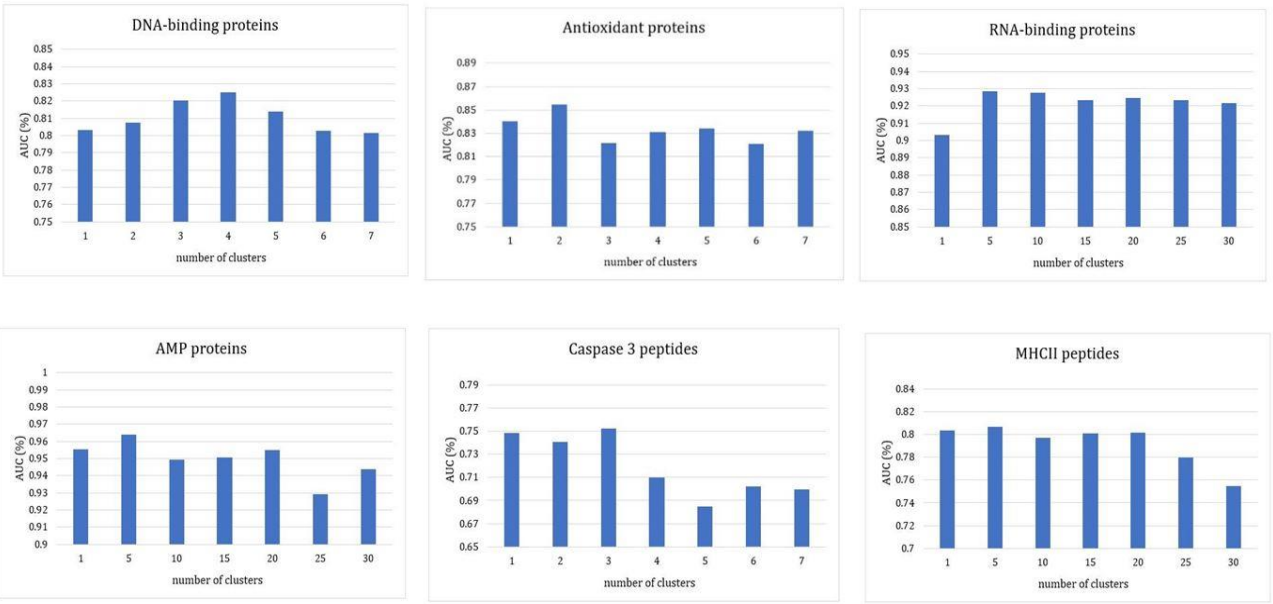

Figure 5. Comparison between AUC values for the FDC and stacking insides clusters at different values of k.

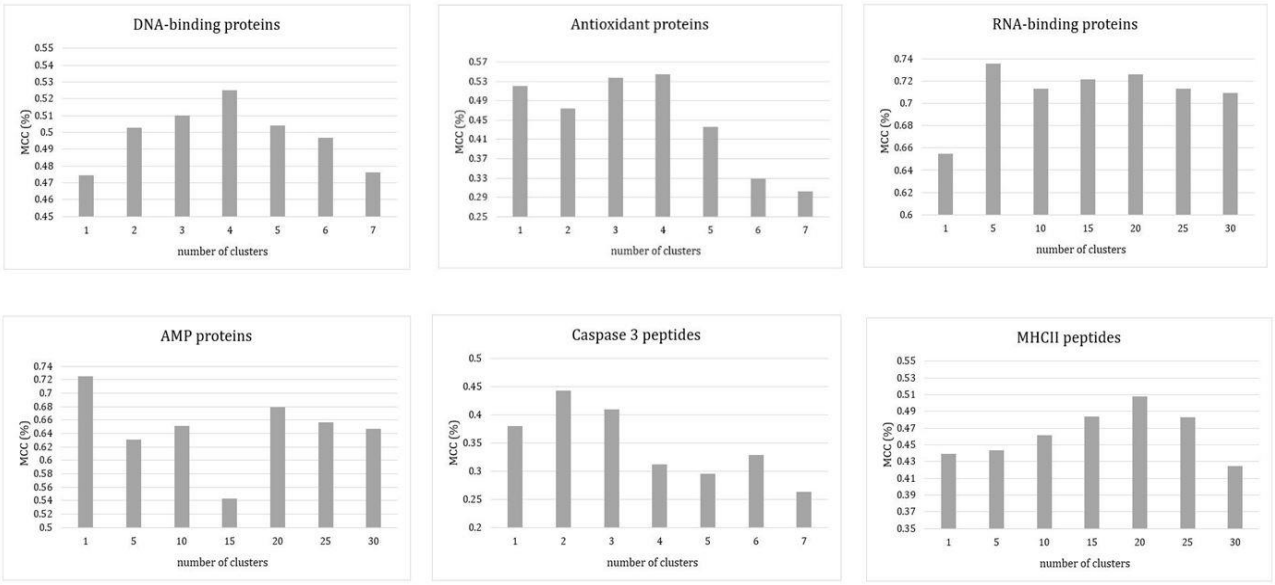

Figure 6. Comparison between MCC values for the FDC and stacking insides clusters at different values of k.

The results on Table II showed that the both approaches improve the performance of the baseline. The main differences between the two approaches that affect performance are:

- The first approach depends on combining the predictions of different values of k, while the second approach depends on finding the best performance at specific value of k. Therefore, the weak clusters may affect the performance of the first approach more than the second approach.
- For the first approach, the meta-classifier was trained on the predictions obtained from the first level for all training data, so all training data share in finding the best hyper-parameter for the meta-classifier, while in the second approach, we trained meta-classifier in each cluster by handling each cluster as a simple problem inside a complex dataset.

We used RF as one of our base classifiers. RF is already an ensemble methods in itself. Our results show that its performance can be improved further by also incorporating structural information extracted in the clustering stage.

VI. Conclusion

We have studied the combined effect of improving protein sequence classifiers with clustering and ensemble learning. Clustering approaches can extract valuable structure information from protein data, while ensembles can stabilize and even boost prediction performance given a diverse set of base classifiers. To this end we have explored two routes. In our first approach we use clustering to generate diverse classifiers for stacking, while in the second approach we apply stacking inside each cluster, which we think of as a homogeneous sub-dataset.

We have evaluated the performance of the proposed approaches on six protein sequence datasets. The performance of the proposed approaches depends on the clusterability of the dataset, the encoding method, and the number of clusters. Our results show that combining structural information of the data obtained by clustering with ensemble classification improves the results in the most cases. We can therefore recommend our methodology for protein function prediction.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Haneen Altartouri and Tobias Glasmachers conceived of the presented approaches. Haneen designed the experiments and carried out the experiments. All authors analyzed the results, and approved the final version of this paper.

References

[1] R. Saidi, M. Maddouri, and E. M. Nguifo, "Protein sequences classification by means of feature extraction with substitution matrices," *BMC Bioinformatics*, vol. 11, p. 175, 2010.
[2] L. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 993-1001, 1990.
[3] J. Kittler, R. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 226-239, 1998.
[4] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in *Proc. the Twenty-First International Conference on Machine Learning*, 2004.
[5] J. Friedman and B. Popescu, "Predictive learning via rule ensembles," *The Annals of Applied Statistics*, vol. 2, 2008.
[6] T. Dietterich, "Ensemble methods in machine learning," in *Proc. International Workshop on Multiple Classifier Systems*, 2000, pp. 1-15.
[7] T. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Mach. Learn.*, vol. 40, 2000.
[8] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123-140, 1996.

[9] F. Yoav and E. S. Robert, "A short introduction to boosting," *Journal of Japanese Society for Artificial Intelligence*, vol. 14, p. 771-780, 1999.
[10] K. Ting and I. Witten, "Stacked generalization: When does it work?" in *Proc. the 15th International Joint Conference on Artificial Intelligence*, Nagoya, Japan, 1997, p. 23-29.
[11] J. Xiao, Y. Tian, L. Xie, and J. Huang, "A hybrid classification framework based on clustering," *IEEE Transactions on Industrial Informatics*, p. 1, 2019.
[12] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in *Proc. SIGCOMM Workshop on Mining Network Data*, 2006, pp. 281-286.
[13] A. Jurek-Loughrey, C. Nugent, Y. Bi, and S. Wu, "Clustering-Based ensemble learning for activity recognition in smart homes," *Sensors (Basel, Switzerland)*, vol. 14, pp. 285-304, 2014.
[14] S. Gaddam, V. Phoha, and K. Balagani, "K-means+id3: A novel method for supervised anomaly detection by cascading k-means clustering and id3 decision tree learning methods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, pp. 345-354, 2007.
[15] R. Rajamohamed and J. Manokaran, "Improved credit card churn prediction based on rough clustering and supervised learning techniques," *Cluster Computing*, vol. 21, pp. 1-13, 2018.
[16] D. Fradkin, "Within-Class and unsupervised clustering improve accuracy and extract local structure for supervised classification," PhD thesis, Rutgers, The State University of New Jersey, 2006.
[17] A. Acharya, E. Hruschka, J. Ghosh, and S. Acharyya, "C3e: A framework for combining ensembles of classifiers and clusterers," in *Proc. the 10th International Conference on Multiple Classifier Systems*, 2011, pp. 269-278.
[18] L. Coletta, E. Hruschka, A. Acharya, and J. Ghosh, "A differential evolution algorithm to optimise the combination of classifier and cluster ensembles," *International Journal of Bio-Inspired Computation*, 2014.
[19] L. Coletta, N. Felix, E. Hruschka, and E. Hruschka, "Combining classification and clustering for tweet sentiment analysis," in *Proc. Brazilian Conference on Intelligent Systems*, 2014.
[20] S. Trivedi, Z. Pardos, and N. Heffernan, "The utility of clustering in prediction tasks," arXiv preprint, arXiv:1509.06163, 2015.
[21] K. C. Chou, "Prediction of protein cellular attributes using pseudo-amino acid composition," *Proteins*, vol. 43, pp. 246-55, 2001.
[22] Y. Fang, Y. Guo, Y. Feng, and M. Li, "Predicting DNA-binding proteins: Approached from chou's pseudo amino acid composition and other specific sequence features," *Amino Acids*, vol. 34, pp. 103-109, 2008.
[23] P. Wang, *et al.*, "Prediction of antimicrobial peptides based on sequence alignment and feature selection methods," *PloS One*, vol. 6, p. e18476, 2011.
[24] K. C. Chou, "Pseudo amino acid composition and its applications in bioinformatics, proteomics and system biology," *Current Proteomics*, vol. 6, pp. 262-274, 2009.
[25] A. Georgiev, "Interpretable numerical descriptors of amino acid space," *Journal of Computational Biology*, vol. 16, no. 5, pp. 703-723, 2009.
[26] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995.
[27] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.
[28] T. Chappell, S. Geva, and J. Hogan, "K-means clustering of biological sequences," in *Proc. 22nd Australasian Document Computing Symposium*, 2017, pp. 1-4.
[29] A. Bustamam, H. Tasman, N. Yuniarti, Frisca, and I. Mursidah, "Application of k-means clustering algorithm in grouping the DNA sequences of hepatitis B virus (HBV)," *AIP Conference Proceedings*, vol. 1862, p. 030134, 07 2017.
[30] J. A. Hartigan and M. A. Wong, "A k-means clustering algorithm," *Applied Statistics*, vol. 28, no. 1, pp. 100-108, 1979.
[31] A. Cucchiara, "Applied logistic regression," *Technometrics*, vol. 34, pp. 358-359, 2012.
[32] M. A. Nielsen, *Neural Networks and Deep Learning*, Determination Press, 2015.
[33] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785-794.
[34] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21-27, 1967.
[35] 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.
[36] W. Z. Lin and D. Xu, "Imbalanced multi-label learning for identifying antimicrobial peptides and their functional types," *Bioinformatics*, vol. 32, p. btw560, 2016.
[37] M. Nielsen and O. Lund, "NN-align. an artificial neural network-based alignment algorithm for MHC class II peptide binding prediction," *BMC Bioinformatics*, vol. 10, p. 296, 2009.
[38] S. Chowdhury, S. Shatabda, and I. A. Dehzangi, "iDNAProt-ES: Identification of DNA-binding proteins using evolutionary and structural features," *Scientific Reports*, vol. 7, 2017.
[39] P. M. Feng, H. Lin, and W. Chen, "Identification of antioxidants from sequence information using naïve bayes," *Computational and Mathematical Methods in Medicine*, vol. 2013, p. 567529, 2013.
[40] X. Zhang and S. Liu, "RBPPred: Predicting RNA-binding proteins from sequence using SVM," *Bioinformatics*, vol. 33, p. 854-862, 2016.
[41] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochimica et Biophysica Acta (BBA) - Protein Structure*, vol. 405, no. 2, pp. 442-451, 1975.
[42] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, pp. 861-874, 2006.

Haneen Altartouri received her master degree in informatics from Palestine Polytechnic University, Palestine, in 2013. Currently, she is pursuing the Ph.D. degree at the Ruhr-University Bochum, Germany, under the supervision of Prof. Tobias Glasmachers. Her main research interests include machine learning and bioinformatics.

Tobias Glasmachers received his Diploma and Doctorate degrees in mathematics from the Ruhr-University of Bochum, Germany, in 2004 and 2008.
He joined the Swiss AI lab IDSIA from 2009 to 2011. Then he returned to Bochum, where he was a junior professor for machine learning at the Institute for Neural Computation (INI) from 2012 to 2018. In 2018 he was promoted to a full professor. His research interests are machine learning and optimization.