Appendix

November 8, 2020

A Feature Encoding Methods

We implement the our method with AAC, AAP and LBE protein sequence-based feature encoding methods. All methods extract fixed length feature vector that is independent from length of the protein sequence. The feature vector used for the interaction is obtained by concatenating the features of each protein.

AAC calculates the frequency of each amino acid and then encapsulates these values in a vector of 20 dimensions [1]. Below, we explain how we compute the AAC feature vector of a given protein. Let us denote the number of copies of the ith amino acid in the given protein with n_i and the total number of amino acids in the given protein with n (i.e., $n = \sum_i n_i$). We define the feature vector of the given protein sequence as follow:

$$\left[\frac{n_1}{n}, \frac{n_2}{n}, \ldots, \frac{n_i}{n}\right] \tag{1}$$

AAP is developed by Chen et al [3]. We can explain the implementation of this method in three steps. In the first step, we decompose the peptides continuously and describe the occurrence of them. For example, MTAEEMK can be decomposed into 6 aaps: MT, TA, AE, EE, EM, MK. Since there are 20 amino acids, the length of feature vector of AAP will be 400 (20x20). In the second step, we find the occurrence frequencies of a given AAP in the proteins with specified functions and other proteins as f^{+}_{AAP} and f^{-}_{AAP}, respectively. Then the AAP scale is centralized and normalized as follow :

$$R = \log\left(\frac{f^{+}_{AAP}}{f^{-}_{AAP}}\right) \tag{2}$$

Let R_{min} and R_{max} represent the minimum and maximum frequency of amino acid pairs. The AAP scale is normalized as follow.

$$R_{AAP} = 2\left(\frac{R - R_{min}}{R - R_{max}}\right) - 1 \tag{3}$$

Finally, we produce a 400-dimensional feature vector by multiplying the occurrence of dipeptides in sequence and the R_{AAP} vector.

LBE is developed by Kosesoy et al [8]. LBE generally consists of three steps. The first step is to divide a given protein sequence into its sub-sequences. The number of sub-sequences, L, must be determined in advance. The length of each sub-sequence, d_i, is obtained according to the equation 4, where N is the length of a protein sequence, and $i = 1, \ldots, L$. d_i is set to the greatest integer that is less than or equal to the output of the right side of the equation 4.

$$d_i = \lfloor i * \frac{N}{L} \rfloor \tag{4}$$

In the second step, the feature vector for a given sequence is extracted. Let us denote the set of locations of the ith amino acid in a given protein with C_i, where $j \in C_i$ as the locations of individual copies. Let N and n_i denote length of the sequence and the scalar value of ith AA in the feature vector of F, respectively. F for a given sequence is calculated as follows:

$$F(n_i) = \sum_{j \in C_i} \frac{j}{N} \tag{5}$$

In the last step, the first and second steps of the LBE implemented on the reversed version of the sequences as well. The final feature vector is generated by concatenating the results of first two step of the algorithm. For a given amino acid sequence, the length of the final feature vector depend on the parameter L, which is used to determine the number of sub-sequences in step 1.

B Prediction Strategies Tested

We have tested our method with diverse classifiers such as statistical based, decision tree and instance based methods.

Bayes Net (BN), is a graph-based model for representing probabilistic relationships between random variables. BN consists of a set of variables, $V = (A_1, A_2, \ldots, A_N)$ and a set of directed edges E. These networks are directed acyclic graphs $G = (V, E)$ that allow efficient and effective representation of the joint probability distribution over a set of random variables [7]. In a BN, each variable is conditionally independent of its non-descendants graph, given a value of its parents G [9]. The joint distribution of $P(V)$ is the product of all conditions specified in the BN defined as follows:

$$P(A_1, A_2, A_3, \ldots, A_N) = \prod_{i=1}^{N} P(A_i/Pa_i) \tag{6}$$

where $P(A_1, A_2, A_3, \ldots, A_N)$ is the probability of a particular combination of V, and $P(A_i/Pa_i)$ is the conditional distribution of A_i, given Pa_i. The conditional distribution of each variable has a parametric form that can be learned by maximum likelihood estimation.

Naive Bayes (NB) is a Bayes rule based classification algorithm which assumes all the variables are conditionally independent. When X contains n attributes conditionally independent of given Y [9]. The value of assumption will be as follows:

$$P(X_1, \ldots, X_n|Y) = \prod_{i=1}^{n} P(X_i/Y) \tag{7}$$

Assuming in general that Y is any discrete-valued variable and, the attributes $X_i, ..., X_n$ are any discrete or real valued attributes. The aim is to define a classifier that will extract the probability distribution over the possible values of Y for each instance of X that needs to be classified. According to Bayes rule, the expression for the probability Y, will take on kth possible value as fallow :

$$P(Y = y_k|X_1, \ldots, X_n) = \frac{P(Y = y_k) \prod_i P(X_i|Y = y_k)}{\sum_j P(Y = y_j) \prod_i P(X_i|Y = y_k)} \tag{8}$$

The equation 8 is the fundamental equation for the NB classifier. If most probable value of Y is to be found for a new instance $X_new = [X_1, .., X_n]$ then the NB classification rule :

$$y \leftarrow \underset{y_k}{argmax} \frac{P(Y = y_k) \prod_i P(X_i|Y = y_k)}{\sum_j P(Y = y_j) \prod_i P(X_i|Y = y_k)} \tag{9}$$

Random Forest (RF) is proposed by Leo Breiman in the 2001 [2]. This algorithm is one of the ensemble methods which based on combination of a large set of decision trees. A random set of variable from training set using to train each tree. Three training parameters needs to be defined in the RF algorithm : n, the number of bootstrap samples; m, the number of different predictors tested at each node; and a node size, the minimal size of the terminal nodes of trees [10]. RF is fast and easy to implement, produce highly accurate predictions and can handle a very large number of input variables without overfitting.

J48 is an implementation of the C4.5 algorithm which is an extension of ID3. This algorithm generates decision trees from set of training data for the prediction of the target variable. At each node of the tree chooses the feature of the data which best splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain. The attribute with the highest gain is chosen to make decision. This process uses the entropy which is a measure of the data disorder. The entropy of $\vec{y}$ is calculated by

$$Entropy(\vec{y}) = \sum_{j=1}^{n} \frac{|y_i|}{|\vec{y}|} \log\left(\frac{|y_i|}{|\vec{y}|}\right) \tag{10}$$

$$Entropy(j|\vec{y}) = \frac{|y_j|}{|\vec{y}|} \log\left(\frac{|y_j|}{|\vec{y}|}\right) \tag{11}$$

and gain is :

$$Gain(\vec{y}, j) = Entropy(\vec{y} - Entropy(j|\vec{y}))$$

The aim is to maximize the gain, dividing by overall entropy due to split argument $\vec{y}$ by value j.

kNN is a one of the most important supervised and non-parametric learning algorithm [5]. The k-NN classification finds a group of k objects in the training set that are closest to the test object and bases the assignment of a label on the predominance of a particular class in this neighborhood [6]. There are three key elements of this approach: a set of labeled objects, a distance or similarity metric to compute distance between objects and the value of k, the number of nearest neighbors(NNs). To classify an unlabeled object, the distance of this object to the labeled objects is computed, its k-NNs are identified and the class labels of these NNs are then used to determine the class label of the object [11] . In our study the k value determined as 3 in all experiments.

K-star is an instance-based classifier. The class of a test instance is based on the class of those instances in the training set similar to it, as determined by some similarity measurement. The underlying assumption of this type of classifiers is that similar cases belong to the same class [4], [12].

C Standard Deviation Results

	ENM					PHI				
Feature	Method	accuracy	precision	recall	f1	mcc	accuracy	precision	recall	f1	mcc
AAC	BN	1.092	0.011	0.011	0.011	0.026	1.471	0.012	0.015	0.013	0.041
	NB	0.953	0.008	0.010	0.009	0.019	1.493	0.009	0.015	0.013	0.029
	kNN	1.488	0.015	0.015	0.014	0.035	1.339	0.008	0.013	0.011	0.029
	K*	0.809	0.005	0.008	0.007	0.013	1.135	0.007	0.011	0.009	0.021
	j48	0.627	0.006	0.006	0.006	0.016	0.758	0.008	0.008	0.008	0.031
	RF	0.707	0.007	0.007	0.008	0.020	0.472	0.010	0.005	0.005	0.027
AAP	BN	1.130	0.011	0.011	0.011	0.027	1.228	0.009	0.012	0.011	0.030
	NB	0.643	0.008	0.006	0.007	0.019	1.008	0.009	0.010	0.009	0.031
	kNN	0.771	0.008	0.008	0.009	0.021	0.856	0.009	0.009	0.009	0.033
	K*	0.927	0.011	0.008	0.010	0.031	0.831	0.008	0.008	0.008	0.033
	j48	1.206	0.013	0.012	0.012	0.031	1.197	0.010	0.012	0.011	0.037
	RF	1.001	0.010	0.010	0.011	0.027	0.591	0.009	0.006	0.008	0.030
LBE	BN	1.054	0.009	0.011	0.010	0.022	1.474	0.010	0.015	0.013	0.034
	NB	0.706	0.009	0.007	0.008	0.022	1.171	0.011	0.012	0.011	0.038
	kNN	1.186	0.012	0.012	0.012	0.031	0.925	0.010	0.009	0.009	0.034
	K*	0.686	0.006	0.007	0.010	0.020	0.567	0.017	0.006	0.010	0.048
	j48	1.002	0.011	0.010	0.010	0.027	1.065	0.008	0.011	0.009	0.030
	RF	0.783	0.008	0.008	0.008	0.021	0.660	0.009	0.007	0.008	0.031

Table 1: Standard deviation observed after 10-fold cross validation for bacillus anthracis data set

		accuracy	precision	recall	f1	mcc	accuracy	precision	recall	f1	mcc
AAC	BN	0.811	0.006	0.008	0.007	0.016	1.100	0.008	0.011	0.009	0.027
	NB	0.914	0.007	0.009	0.009	0.018	1.079	0.006	0.011	0.009	0.018
	kNN	1.122	0.011	0.011	0.011	0.026	1.018	0.007	0.010	0.008	0.025
	K*	0.835	0.007	0.008	0.008	0.018	1.089	0.008	0.011	0.009	0.028
	j48	0.934	0.009	0.009	0.009	0.022	0.859	0.010	0.009	0.009	0.036
	RF	0.583	0.005	0.006	0.007	0.016	0.450	0.004	0.005	0.008	0.026
AAP	BN	0.877	0.007	0.009	0.008	0.018	0.877	0.007	0.009	0.008	0.018
	NB	0.918	0.011	0.009	0.009	0.026	0.918	0.011	0.009	0.009	0.026
	kNN	0.691	0.007	0.007	0.008	0.020	0.691	0.007	0.007	0.008	0.020
	K*	0.832	0.011	0.009	0.010	0.031	0.832	0.011	0.009	0.010	0.031
	j48	1.274	0.012	0.013	0.012	0.030	1.274	0.012	0.013	0.012	0.030
	RF	0.665	0.007	0.007	0.007	0.019	0.665	0.007	0.007	0.007	0.019
LBE	BN	1.185	0.010	0.012	0.011	0.025	1.185	0.010	0.012	0.011	0.025
	NB	0.916	0.011	0.009	0.010	0.027	0.916	0.011	0.009	0.010	0.027
	kNN	0.855	0.009	0.009	0.009	0.023	0.855	0.009	0.009	0.009	0.023
	K*	0.541	0.006	0.005	0.007	0.017	0.541	0.006	0.005	0.007	0.017
	j48	0.552	0.006	0.006	0.006	0.015	0.552	0.006	0.006	0.006	0.015
	RF	0.863	0.009	0.009	0.009	0.024	0.863	0.009	0.009	0.009	0.024

Table 2: Standard deviation results observed after 10-fold cross validation for yersinia pestis data set

MERGED						
		accuracy	precision	recall	f1	mcc
AAC	BN	1.105	0.006	0.011	0.009	0.022
	NB	1.432	0.008	0.014	0.012	0.025
	kNN	0.877	0.008	0.009	0.008	0.027
	K*	0.915	0.006	0.009	0.008	0.019
	j48	0.856	0.010	0.009	0.008	0.035
	RF	0.331	0.004	0.003	0.005	0.019
AAP	BN	0.627	0.004	0.006	0.005	0.013
	NB	0.620	0.006	0.006	0.006	0.021
	kNN	0.596	0.007	0.006	0.006	0.025
	K*	0.559	0.005	0.005	0.005	0.021
	j48	0.695	0.006	0.007	0.007	0.023
	RF	0.346	0.004	0.003	0.005	0.019
LBE	BN	1.006	0.005	0.010	0.008	0.018
	NB	0.750	0.006	0.008	0.006	0.022
	kNN	0.581	0.006	0.006	0.006	0.022
	K*	0.349	0.009	0.003	0.006	0.028
	j48	0.752	0.007	0.008	0.007	0.026
	RF	0.458	0.006	0.005	0.006	0.023

Table 3: Standard deviation results observed after 10-fold cross validation for merged data set

References

[1] Manoj Bhasin and Gajendra PS Raghava. “Classification of nuclear receptors based on amino acid composition and dipeptide composition”. In: *Journal of Biological Chemistry* 279.22 (2004), pp. 23262–23266.

[2] Leo Breiman. “Random forests”. In: *Machine learning* 45.1 (2001), pp. 5–32.

[3] J Chen et al. “Prediction of linear B-cell epitopes using amino acid pair antigenicity scale”. In: *Amino acids* 33.3 (2007), pp. 423–428.

[4] John G Cleary, Leonard E Trigg, et al. “K*: An instance-based learner using an entropic distance measure”. In: *Proceedings of the 12th International Conference on Machine learning*. Vol. 5. 1995, pp. 108–114.

[5] Belur V Dasarathy. “Nearest neighbor ({NN}) norms:{NN} pattern classification techniques”. In: (1991).

[6] Evelyn Fix and Joseph L Hodges Jr. *Discriminatory analysis-nonparametric discrimination: consistency properties*. Tech. rep. California Univ Berkeley, 1951.

[7] Nir Friedman, Dan Geiger, and Moises Goldszmidt. “Bayesian network classifiers”. In: *Machine learning* 29.2-3 (1997), pp. 131–163.

[8] Irfan Kosesoy, Murat Gök, and Cemil Öz. “A new sequence based encoding for prediction of host–pathogen protein interactions”. In: *Computational Biology and Chemistry* 78 (2019), pp. 170–177. ISSN: 1476-9271. DOI: `https://doi.org/10.1016/j.compbiolchem.2018.12.001`. URL: `http://www.sciencedirect.com/science/article/pii/S1476927117308848`.

[9] V Muralidharan and V Sugumaran. “A comparative study of Naıve Bayes classifier and Bayes net classifier for fault diagnosis of monoblock centrifugal pump using wavelet analysis”. In: *Applied Soft Computing* 12.8 (2012), pp. 2023–2029.

[10] Simone Vincenzi et al. “Application of a Random Forest algorithm to predict spatial distribution of the potential yield of Ruditapes philippinarum in the Venice lagoon, Italy”. In: *Ecological Modelling* 222.8 (2011), pp. 1471–1478. ISSN: 0304-3800. DOI: `http://dx.doi.org/10.1016/j.ecolmodel.2011.02.007`. URL: `http://www.sciencedirect.com/science/article/pii/S0304380011000640`.

[11] Xindong Wu et al. “Top 10 algorithms in data mining”. In: *Knowledge and information systems* 14.1 (2008), pp. 1–37.

[12] Du Zhang. *Advances in machine learning applications in software engineering*. IGI Global, 2006.