High performance set of PseAAC and sequence based descriptors for protein classification

Loris Nanni, Sheryl Brahnam, Alessandra Lumini

www.elsevier.com/locate/yjtbi

High performance set of PseAAC and sequence based descriptors for protein classification

Loris Nanni, Sheryl Brahnam, Alessandra Lumini

PII: S0022-5193(10)00290-0
DOI: doi:10.1016/j.jtbi.2010.06.006
Reference: YJTBI6026

To appear in: *Journal of Theoretical Biology*

Received date: 1 March 2010
Revised date: 31 May 2010
Accepted date: 2 June 2010

Cite this article as: Loris Nanni, Sheryl Brahnam and Alessandra Lumini, High performance set of PseAAC and sequence based descriptors for protein classification, *Journal of Theoretical Biology*, doi:10.1016/j.jtbi.2010.06.006

High performance set of PseAAC and sequence based descriptors for protein classification

Loris Nanni[a 1], Sheryl Brahnam[b], Alessandra Lumini[a]

[a] Department of Electronic, Informatics and Systems (DEIS), Università di Bologna,
Via Venezia 52, 47023 Cesena, Italy.
{loris.nanni, alessandra.lumini}@unibo.it

[b] Computer Information Systems, Missouri State University,
901 S. National, Springfield, MO 65804, USA
sbrahnam@missouristate.edu

Abstract

The study of reliable automatic systems for protein classification is important for several domains, including finding novel drugs and vaccines. The last decade has seen a number of advances in the development of reliable systems for classifying proteins. Of particular interest has been the exploration of new methods for extracting features from a protein that enhance classification for a given problem. Most methods developed to date, however, have been evaluated in only one or two application areas. Methods have not been explored that generalize well across a number of applications areas and datasets. The aim of this study is to find a general method, or an ensemble of methods, that work well on different protein classification datasets and problems.

Towards this end, we evaluate several feature extraction approaches for representing proteins starting from their amino acid sequence as well as different feature descriptor combinations using an ensemble of classifiers (support vector machines). In our experiments, more than ten different protein descriptors are compared using nine different datasets. We develop our system using a blind testing protocol, where the parameters of the system are optimized using one dataset and then validated using the other datasets (and so on for each dataset). Although different stand-alone classifiers work well on some datasets and not on others, we have discovered that fusion among

[1] corresponding author: Tel.: +39 0547 339121; fax: +39 0547 338890.

different methods obtains a good performance across all the tested datasets, especially when using the weighted sum rule.

Included in our feature descriptor combinations is the introduction of two new descriptors, one based on wavelets and the other based on amino acid groups. Using our system, both outperform their standard implementations. We also consider as a baseline the simple amino acid composition (AC) and dipeptide composition (2G), since they have been widely used for protein classification. Our proposed method outperforms AC and 2G.

Keywords: proteins classification; machine learning; ensemble of classifiers, support vector machines.

1. Introduction

Extracting features from proteins for protein classification has value in many applications, including subcellular localization (Chou and Shen, 2007a; Chou and Shen, 2010; Shen and Chou, 2010) and protein-protein interactions (Nanni and Lumini, 2006). Several techniques for extracting features from proteins have been developed (Chou and Shen, 2007a). They can be roughly classified, according to their characteristics, into three main classes. The first class comprises the Chou's pseudo amino acid (PseAA) composition (Chou and Shen, 2007a), probably the most used feature extractor for proteins, and its variants. To avoid losing important information hidden in protein sequences, the pseudo amino acid composition (PseAAC) was proposed (Chou, 2001; Chou, 2005) to replace the simple amino acid composition (AAC) for representing the sample of a protein. PseAA composition represents a protein sequence with a discrete model without completely losing its sequence order information; the model is composed of a set of more than 20 discrete factors, where the first 20 factors represent the components of its conventional amino acid (AA) composition while the additional factors incorporate some of its sequence order information via

various modes (i.e., as a series of rank-different correlation factors along a protein chain). For a summary about its development and applications, such as how to use the concept of Chou's PseAAC to develop 16 different forms of PseAAC, including those that are able to incorporate the functional domain information, GO (gene ontology) information, Cellular Automaton image information, sequential evolution information, among many others, see a recent comprehensive review (Chou, 2009). In this paper, we consider also other additional forms of PseAAC in the hopes of further strengthening the power of PseAAC.

The second class includes feature extractors based on a vectorial representation of the protein where the feature extraction is not explicitly related to groups of amino acids. For example, in (Nanni and Lumini, 2006) the physicochemical encoding is proposed, it combines the value of a given property for an amino acid together with its 2-grams representation.

The third class includes methods based on kernels. One of the first approaches is the Fisher kernel (Jaakkola et al., 1999) proposed for remote homology detection. A different kernel, the mismatch string kernel, is proposed in (Leslie et al., 2004), which measures the sequence similarity based on shared occurrences of subsequences. In (Leslie et al., 2004) it is shown that string kernels perform similarly to Fisher kernel but with a lower computational cost. A class of new kernels is developed in (Lei and Dai, 2005) for vectors derived from k-peptide vectors mapped by a matrix of high-scored pairs, measured by BLOSUM62 scores, of k-peptides: the kernel functions are used for training a support vector machine and their good performance for predicting protein subcellular localization is reported in (Lei and Dai, 2005). Another interesting approach is the bio-basis function neural network (Yang and Thomson, 2005). In this method the sequences are not encoded in a feature space but rather the distances obtained by sequence alignment are used to train the neural network.

In practical applications, particularly in developing high throughput tools for predicting various biological attributes, many different pseudo amino acid compositions for biological sequence feature representation have been developed and widely used. For example, cellular

automata image (Lin et al., 2009; Xiao et al., 2008a; Xiao et al., 2009a; Xiao et al., 2006a), complexity measure factor (Xiao et al., 2006b; Xiao et al., 2005); Grey dynamic model (Xiao and Lin, 2009; Xiao et al., 2008b); functional domain composition (Xiao et al., 2009b).

The aim of this work is to propose a general system for protein classification based on the combinations of different feature extractors, mainly derived from the first two classes, and to evaluate the system on different protein classification problems using different datasets. Several studies have proposed systems that work well on a given dataset, but their parameters tuning (e.g., the number of psychochemical properties) are optimal only for the proposed problem and not for others. Methods have not been explored that generalize well across a number of applications areas and datasets. The aim of this study is to find a general method, or ensemble, that works well on different protein classification datasets and problems.

Some advantages in exploring protein classification methods that generalize well include deepening our understanding of protein representation, speeding up real world development in new areas involving protein classification, developing more robust and powerful classification systems, and providing standards for comparing protein classification methods across a host of application areas.

To obtain an ensemble of methods that works well on different protein problems, we study several combinations of feature extractors, and we perform an experimental evaluation on seven different datasets and nine different test sets. As a result of our experiments, we obtain a number of statistically robust observations regarding the effectiveness of the proposed system.

The remaining of the paper is organized as follows. In section 2 we introduce the feature extraction methods explored in this work. In section 3 we report experimental results obtained on nine different classification problems. Finally, in section 4, we draw a number of conclusions.

2 Feature Extraction

Several studies, e.g., (Chou and Shen, 2007c; Nanni and Lumini, 2008), deal with the problem of finding a compact and effective representation from proteins, because there are many classification problems (e.g., subcellular localization, protein-protein interactions) that require a machine learning approach. In many cases a feasible solution is based on extracting a fixed length encoding to be coupled with a general purpose classifier. In this section we briefly describe the encoding methods explored in this study.

For each feature extraction method we used the support vector machine[2] (SVM) as the classifier. The SVM is a technique for classification that arose from the field of statistical learning theory (Cristianini and Shawe-Taylor, 2000). SVM is a binary-class prediction method trained to find the equation of a hyperplane that divides the training set leaving all the points of the same class on the same side while maximizing the distance between the two classes and the hyperplane. In cases where a linear decision boundary does not exist, a kernel function can be used. A kernel function projects the data onto a higher-dimensional feature space where they can be separated by a hyperplane. Typical kernels are polynomial kernels and the radial basis function kernel. All the features used for training SVM are linearly normalized to [0 1] considering the training data.

The set of physicochemical properties are obtained from the amino acid index (Kawashima and Kanehisa, 2000) database[3]. An amino acid index is a set of 20 numerical values representing the different physicochemical properties of amino acids. This database currently contains 544 indices and 94 substitution matrices. Unfortunately, many properties are highly correlated with each other. To reduce the number of properties considered in the feature extraction process, we could select the *k* best physicochemical properties for a particular classification problem by running Sequential Forward Floating Selection (SFFS)[4] (where the features are the physicochemical properties) as in

[2] SVM is implemented as in the OSU svm toolbox

[3] available at http://www.genome.jp/dbget/aaindex.html. We have not considered the properties where the amino acids have value 0 or 1.

[4] implemented as in PRTools 3.1.7 Matlab Toolbox

our previous work, e.g., (Nanni and Lumini, 2006). We noticed when running some new tests that a random selection of a large number of properties works as well as a smaller number of properties specifically selected by SFFS. It is thus a viable option to use either the ten best selected properties selected by SFFS or the 50 random selected properties. By using a randomly selected set of properties, we avoid a parameter, i.e., we no longer need to an elaborate method to select a set of properties for a given problem. Although it is the case that using the randomly selected properties will require more computational power than using those selected by SFFS (50 versus 10 features in the feature set), we obtain the advantage of producing, as will be noted in the experimental section, a more robust generic system. It should be noted that for each of the protein descriptors, where the features are extracted considering a given physiochemical property, we combine the score obtained from the 50 randomly selected properties using the sum rule.

Since the aim of this work is to find an ensemble of methods that works well on different problems, in each protein classification problem, we extract a set of features using the descriptors described below. We then use a leave-one-out dataset testing protocol for selecting (using SFFS) a set of *n* descriptors used to train the SVMs. These classifiers are combined by a fusion rule (Kittler, 1998). We have also tested the sum rule and the weighted sum rule, where a different weight is assigned to each descriptor. The weights of the weighted sum rule are selected by SFFS. The testing protocol we use is blind: the set of descriptors used to classify the proteins in a given problem are selected using the other remaining datasets.

SFFS is a bottom up search procedure introduced by (Pudil et al., 1994). It uses a forward step followed by a conditional backward step. The forward step starts from an initially empty set of features and successively adds features from a set of original candidates in order to optimize a given objective function. Each time a single feature is added, a backward step is performed that identifies the least significant feature in the current feature set and removes it, unless it is the last property added. The number *k* of retained features is determined according to the objective function, as the minimum number of features that maximizes the performance.

2.1 Physicochemical 2-Grams (*P2G*)

The physicochemical 2-grams (Nanni and Lumini, 2006) are a model for protein representation that combines the value of a given property for an amino acid together with its 2-grams representation (2G). 2G is a vector of 20^2 values c_l, each counting the number of occurrences of a given couple of amino acids in a protein sequence. We define $F^d(i,j)$ for a given physiochemical property d, and a couple of amino acids i,j ($i,j \in [1,..20]$) as:

$$F^d(i,j) = \left(\frac{h(i,j)\cdot index(i,d)}{Len-1}, \frac{h(i,j)\cdot index(j,d)}{Len-1}\right) \quad i,j \epsilon [1..20]$$

where i and j denote the 20 different amino acids[5]; *Len* is the length of the protein; d denotes the selected physicochemical property; the function *index*(i,d) return the value of the property d for the amino acid i; the function $h(i, j)$ count the number of occurrences of a given couple of amino acids (i, j) in a protein sequence.

The feature vector of a protein for a given physiochemical property d is made by the concatenation of all the $F^d(i,j)$ for i,j=1...20; Therefore, we obtain a 800-dimensional vector.

2.2 Quasi Residue Couple (*RC*)

The quasi residue couple is a model for protein representation proposed by (Nanni, 2006) and inspired by Chou's quasi-sequence-order model and Yuan's Markov chain model (Guo et al., 2005). This encoding combines the information related to a fixed physicochemical property of the protein with the sequence order effect of the composition of the amino acid. A residue couple model of order less than three (Guo et al., 2005) is considered to represent the sequence. Each nonzero entry in the residue couple is substituted by the corresponding value of the selected property.

In this work we use the residue couple model with order $m \leq 3$, which for a physicochemical property d is given by:

[5] we use the indexes 1,2,…,20 to represent the 20 native amino acids, respectively, corresponding to the alphabetical order of their single letter codes: A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, and Y.

$$P_m^d(i,j) = \frac{1}{Len-m}\sum_{n=1}^{Len-m} H_{i,j}(n, n+m, d) + H_{j,i}(n+m, n, d) \quad i,j\epsilon[1..20]$$

where *i* and *j* denote the 20 different amino acids; *Len* is the length of the protein; *d* denotes the selected physicochemical property; the function *index*(*i*,*d*) returns the value of the property *d* for the amino acid *i*; and the function $H_{i,j}(a, b, d)$ = *index*(*i*,*d*), if the amino acid in location *a* is *i* and the one in location *b* is *j* and is 0 otherwise.

The parameter *m* is called the order of the residue couple model and the feature vector that describes a given protein is a 400-dimensional vector obtained by calculating $P_m^d(i,j)$ for each couple of *i j*. In the present work, we extract the *RC* features for *m* ranging from 1 to 3, and we concatenate the resulting descriptors into a 1200-dimensional vector.

2.3 Autocovariance approach (*AUC*)

In (Zeng et al., 2009) a sequence-based algorithm combining the augmented Chou's pseudo amino acid composition (Chou, 2001) based on auto covariance is presented. In (Chou, 2001), a set of pseudo-amino-acid-based features[6] are extracted from a given protein as the concatenation of the 20 standard amino acid composition values (AC) and *m* values reflecting the effect of sequence order (where *m* is a parameter denoting the maximum distance between two considered amino acids *i*,*j*).

In this work we consider only the last *m* feature of the vector $\boldsymbol{C}$ = ($C_1,\ldots,C_{20}$, $C_{20+1}^d,\ldots, C_{20+m}^d$) which is defined as:

$$C_{20+l}^d = \sum_{k=1}^{Len-l} \frac{(index(A(k),d) - M_d)\cdot(index(A(k+l),d) - M_d)}{V_d \cdot (Len-l)} \quad l\epsilon[1..m]$$

where $A(k)$ denotes the index of the amino acid in the k^{th} position of the protein; *Len* is the length of the protein; *d* denotes the selected physicochemical property; the function *index*(*i*,*d*) returns the

[6] Extracted by the matlab code shared by the original authors.

value of the property *d* for the amino acid *i*; and M_d and V_d are normalization factors denoting the average and the variance of the physicochemical property *d* on the 20 amino acids.

$$M_d = \frac{1}{20}\sum_{i=1}^{20} index(i,d)$$

$$V_d = \frac{1}{20}\sum_{i=1}^{20} (index(i,d) - M_d)^2$$

In this work we set m_{max}=20. We perform the feature extraction for each m=1...m_{max} for each physicochemical protein present in the Amino Acid index database (Kawashima and Kanehisa, 2000). Therefore, the number of extracted features is given by $np \times m_{max}$, where *np* equals the number of physicochemical properties.

2.4 Amino Acid Group Based Physicochemical Encoding (*AAG*)

This method, proposed in (Hu & Zhang, 2009), is based on clustering amino acids considering the value of a given physicochemical property *d*. Given a protein sequence *A(k)* k=1...*Len*, its index score vector S^d is obtained by replacing each amino acid with its physicochemical value:

$$S^d(k) = index(A(k), d) \quad k = 1 \dots Len$$

The vector S^d is threshold bounded according to the value of its elements:

$$T^d(k) = \begin{cases} 1 & S^d(k) \geq M_d + \lambda\sqrt{V_d} \\ 0 & otherwise \\ -1 & S^d(k) \leq M_d + \lambda\sqrt{V_d} \end{cases}$$

where M_d and V_d are normalization factors denoting the average and the variance of the physicochemical property *d* on the 20 amino acids; λ is a parameter fixed to 1.

The clustering procedure consists in merging adjacent amino acids with equal positive or negative labels into an amino acid group (AAG). Then each group is represented by a couple of values denoting the initial position and the length of the group. Only the AAGs having a minimum

length L (L=2 in this work) are considered. To give some tolerance to small gaps, we will keep merging two positive or negative AAGs if the gap between them is 1 (in this work an element k can be considered a gap only if $T^d(k) = 0$). For example the sequence [-1;0;-1;0;0;0;0;0;1;0;1;1] has a positive AAG group of length 4 in position 8. In (Hu & Zhang, 2009) only the first AAG group of positive amino acids is considered. In this work we test the following configurations:

- *PA*, only the first AAG group of positive amino acids is considered for each physicochemical property (the final vector has length 2×*np, np* = number of physicochemical properties);
- *NA*, only the first AAG group of negative amino acids is considered for each physicochemical property (the final vector has length 2×*np, np* = number of physicochemical properties);
- *PN*, the features are the number (normalized with the length of the protein) of positive and negative AAG groups for each physicochemical property (the final vector has length 2×*np, np* = number of physicochemical properties).

The features after the computation are normalized considering the length of the protein.

Due to the large number of features extracted with this method, instead of using SVM as the classifier, we use a random subspace of SVM. Random subspace (RS) (Ho, 1998) is a method for creating ensembles that modifies the training data set, generating K new training sets containing only a subset of the lower dimensionality of the original features. Then the scores of the classifiers trained on these modified training sets are combined by sum rule. In this work we design an ensemble of K=50 SVM classifiers, and we generate feature vectors containing only a random set of 50% of all the features.

2.5 AAIndexLoc (AA)

The AAIndexLoc (Tantoso and Li, 2008) describes a given protein **P** as follows:

- ***Amino acid (AC) composition (20 features)***: this is fraction of a given amino acid in **P**;
- ***Weighted AA composition (20 features)***: this is defined for a given amino acid *y* as (Amino acid composition of *y*)×(index value *a* for the amino acid *y*);
- ***Five-level grouping composition (25 features)***: this is where the amino acids are classified by k-means clustering into five groups considering their amino acid index values. The five-level dipeptide composition is then performed. The five-level dipeptide composition is defined as the composition of the occurrence of two consecutive groups, see (Tantoso and Li, 2008) for more details.

2.6 Global Encoding (GE)

In this method, proposed in (Xi et al., 2009), the amino acids are first classified into the following six classes:

A1 = {A,V,L,I,M,C}

A2 = {F,W,Y,H}

A3 = {S,T,N,Q}

A4 = {K,R}

A5 = {D,E}

A6 = {G,P}

For each combination, each of which contains three different classes (hence we have ten combinations), we extract a different feature set.

For example, a given combination could be: {A1, A2, A3} vs {A4, A5, A6}. The protein vector is then transformed into a numerical sequence where a given amino acid is assigned a value of 1 if it belongs to {A1, A2, A3}, otherwise it is assigned a value of 0. The first set of extracted features is the “composition," i.e., the frequency of 0s and 1s. The latter set of extracted features is

the "transition", i.e., the percent of frequency with which 1 is followed by 0 or 0 is followed by 1 in a characteristic sequence.

Moreover, we have tested a modified GE where the amino acids are classified into different classes using a genetic algorithm (GA) approach proposed in (Nanni and Lumini, 2008). In this study, we run the GA *n* times and then combine these *n* results.

2.7 Full Sequence (FS)

This protein descriptor is based on all the physicochemical properties of the AAindex. In order to represent a protein sequence, a single feature is extracted for each physicochemical property. The average value of that physicochemical property of the amino acids of that protein is defined as:

$$\sum_{i=1}^{20} A_d(i) fr(i)$$

where $A_d(i)$ is the value of the *i*-th amino acid of the *d*-th physicochemical property and $fr(i)$ is the compositional fraction of the *i*-th amino acid.

2.8 N-Gram (NG)

NG is similar to the standard 2G but here we train five different SVMs. Each classifier is trained using a different *N*-peptide composition with different amino acid alphabets. We have used the following alphabets (Murphy et al., 2000):

G–I–V–F–Y–W–A–L–M–E–Q–R–K–P–N–D–H–S–T–C

LVIM–C– A–G– S–T–P–FY–W–E–D–N–Q–KR–H

LVIMC–AG–ST–P–FYW–EDNQ–KR–H

LVIMC–ASGTP–FYW–EDNQ–KRH

LVIMC–ASGTP–FYW–EDNQKRH

From the first two alphabets, we extract the 2-grams. From the other three alphabets, we extract the 3-grams. These five classifiers are combined by weighted sum rule, where the weight of the first method is 1, the fourth is 0.5, and the last (which is the SVM trained with the reduced alphabets with five elements) is 0.25.

Moreover, we tested a modified NG where the amino acids are classified into different classes using a GA approach proposed in (Nanni and Lumini, 2008). In this study we run the GA ten times, and then combine the ten results. We assign a weight of 10 when the standard alphabets is coupled with 2-gram, and a weight of 1 in the others cases (i.e., the alphabets obtained by GA). All these weights are calculated using only the training data.

2.9 Wavelet descriptor (WA)

Recently some studies have shown that it is possible to extract features from the protein using wavelets. First the protein sequence is converted to a numerical sequence, substituting each amino acid with its value of a given physicochemical property. In (Li et al., 2008) the Meyer continuous wavelet is applied to the numerical sequence. Then the wavelet power spectrum is extracted considering different decomposition scales. We obtain the best performance with 100 decomposition scales and without the applications of the principal component analysis as in (Li et al., 2008).

In this work, we test a different approach for extracting features from the continuous Meyer wavelet image obtained by a given protein: we extract the dominant local ternary patterns (DLTP). DLTP is a combination of dominant local binary patterns and local ternary patterns. Dominant local binary pattern (DLP) was proposed in (Liao et al., 2009) for selecting the rotation invariant patterns to be selected in local binary pattern (LBP). Instead of selecting the uniform patterns, they proposed choosing those patterns that represent 80% of the whole pattern occurrences in the training data. The LBP operator is calculated by evaluating the binary differences between the gray value of a pixel **x** and the gray values of P neighboring pixels on a circle of radius R around **x**. The LBP

operator is made rotation invariant by selecting the smallest value of P-1 bitwise shift operations on the binary pattern. A pattern is considered uniform if the number of transactions in the sequence between 0 and 1 is less than or equal to two. In local ternary patterns (LTP) (Tan and Triggs, 2007) the difference between a pixel **x** and its neighbor **u** is encoded by 3 values according to a threshold τ : 1 if $\mathbf{u} \geq \mathbf{x} + \tau$; -1 if $\mathbf{u} \leq \mathbf{x} - \tau$; else 0. The ternary pattern is then split into two binary patterns by considering its positive and negative components. Finally, the histograms that are computed from the binary patterns are concatenated to form the feature vector. Here τ=0.15; P=16; R=2. We name this method, based on dominant LTP, as DL in section 3, table 4.

In (Qiu et al., 2009) the biorthogonal discrete wavelet is used to describe a protein from the wavelet coefficients. Using different scales, the maximum, minimum, mean and standard deviation values are extracted. We name this method BASE in section 3, table 4. To improve the method proposed in (Qiu et al., 2009), we also propose extracting the first five discrete cosine coefficients from the approximation coefficients, and the maximum, minimum, mean and standard deviation values from both detail and approximation coefficients of the wavelet decomposition (4 scales are used). We name this method DW in section 3, table 4.

Finally, we examine the performance benefit of concatenating DL and DW features, which we name DW+DL.

2.10 Split amino acid composition (SAC)

With SAC (Kumar et al., 2005; Verma et al., 2009) the protein sequence is divided into parts, and the composition of each part is calculated separately. Each protein is divided into three parts: (*i*) 20 amino acids of N-termini, (*ii*) 20 amino acids of C-termini, and (*iii*) the region between these two terminuses.

2.11 Mismatch kernel (MK)[7]

MK (Leslie et al., 2004) is a discriminative approach for the protein classification problem. It measures sequence similarity based on shared occurrences of *k*-length subsequences (in our experiments *k*=3) counted with up to *m* mismatches (in our experiments *m*=1) that do not rely on any generative model for the positive training sequences. We use this method for extracting a fixed length feature vector so that any standard classifier could be used.

3 Experiments

This section reports the results of an experimental evaluation of the protein descriptors performed on several datasets for testing the approaches of protein classification.

3.1 Datasets and testing protocol

The datasets used in this work are described below. We have used the same protocols as reported by the original creators. The performance of the different approaches combined in this work are evaluated and compared with the performance of the stand-alone descriptors. To reduce homology bias, a culling program performed a redundancy cutoff to winnow those sequences which have a given sequence identity to any other protein of the same class. In each dataset the sequence identity threshold used to cutoff the proteins is reported.

The proposed approach has been evaluated on the following datasets:

GPCR (Xiao et al., 2009a): this dataset contains G protein-coupled receptor (GPCR) and non-GPCR. The aim is to identify a query protein as GPCR or non-GPCR. None of the proteins included has ≥ 40% pairwise sequence identity to any other in the same subset.

GRAM (Shen and Chou, 2007b)**:** this dataset contains gram-positive proteins that belong to five subcellular location sites: (1) cell wall, (2) cytoplasm, (3) extracell, (4) periplasm, and (5) plasma membrane. To eliminate redundancy and homology bias, only those proteins that have

[7] Extracted by the matlab code shared by the original authors

<25% sequence identity to any other in a same subcellular location were allowed to be included in the benchmark datasets. The aim is to classify a given query protein in a given localization.

Viral (VIR) (Shen and Chou, 2007a)**:** the subcellular localization of viral proteins within a host cell or virus-infected cell is very useful for studying the function of viral proteins as well as designing antiviral drugs. This dataset contains proteins that belong to: cytoplas, extracellular, nucleus, and plasma membrane. The aim is to classify a given query protein in a given localization. None of the proteins has 25% sequence identity to any other in the same subset (subcellular location).

Membrane sub-cellular (MEM) (Chou and Shen, 2007c)**:** this dataset contains membrane proteins that belong to 8 membrane types: (1) single-pass type I transmembrane, (2) single-pass type II, (3) single-pass type III, (4) single-pass type IV, (5) multipass transmembrane, (6) lipid-chain-anchored membrane, (7) GPI-anchored membrane, and (8) peripheral membrane. The aim is to classify a given query protein in a given localization. None of the proteins has 80% sequence identity to any other in the same subset (subcellular location).

Virulent dataset (Garg and Gupta, 2008)**:** this dataset contains bacterial virulent protein sequences that were retrieved from the SWISS-PROT (Bairoch and Apweiler, 2000) and VFDB, an integrated and comprehensive database of virulence factors of bacterial pathogens, (Chen et al., 2005). It consists of 1025 virulent and 1030 nonvirulent bacterial sequences. It is used as training set, as in the original testing protocol (Garg and Gupta, 2008) for the following three testing sets: ADHESINS dataset, Independent dataset 1, Independent dataset 2.

ADHESINS dataset (VIR1) (Garg and Gupta, 2008): this dataset consists of 469 adhesins and 703 non-adhesins proteins (including several archaebacterial, viral and yeast non virulent proteins).

Independent dataset 1 (VIR2) (Garg and Gupta, 2008): this dataset consists of 83 SWISS-PROT sequences (40 virulent and 43 nonvirulent protein sequences) such that there are no two sequences that are more than 40% similar.

Independent dataset 2 (VIR3) (Garg and Gupta, 2008)**:** this dataset consists of 141 virulent and 143 nonvirulent sequences from bacterial pathogens sequences of organisms that were not represented in the Virulent dataset, divided as follows:

- Campylobacter (39 virulent and 40 nonvirulent protein sequences);
- Neisseria (25 virulent and 24 nonvirulent);
- Bordetella (27 virulent and 27 nonvirulent sequences);
- Haemophilus (35 virulent and 35 nonvirulent);
- Listeria (15 virulent and 17 nonvirulent).

Human protein-protein interaction (HUM) (Bock and Gough, 2003)**:** this dataset contains a total of 1882 human protein pairs. Each pair of proteins is labeled as either an *interacting pair* or a *non-interacting pair*.

Helicobacter protein-protein interaction (HEL) (Bock and Gough, 2003)**:** this dataset contains a total of 2916 helicobacter protein pairs. Each pair of proteins is labeled as either an *interacting pair* or a *non-interacting pair*.

A summary of the characteristics of these datasets is reported in Table 1.

We use the area under the ROC curve (AU)[8] (Fawcett, 2004) as the performance indicator. AU is a scalar measure that can be interpreted as the probability that the classifier will assign a lower score to a randomly picked positive pattern than to a randomly picked negative pattern. When a multiclass dataset is used, the one-versus-all area under ROC curve is used as performance indicator (Landgrebe & Duin, 2007).

The area under the ROC is considered one of the most reliable performance indicators as it is based on both sensitivity and specificity. Accuracy is not as reliable an indicator (Qin, 2006); it is thus not reported.

[8] Implemented as in DDtool 0.95 Matlab Toolbox

DATASET		#E	#C
GPCR proteins	**training**	365	2
	independent	365	
GRAM proteins	**training**	220	5
	independent	232	
Virulent Proteins	**training**	2055	2
	ADHESINS	1172	
	independent 2	83	
	independent 3	284	
Membrane sub-cellular	**training**	3249	8
	independent	4333	
Viral proteins	**training**	70	4
	independent	42	
Human protein-protein	**training**	941	2
	independent	941	
Helicobacter protein-protein	**training**	1458	2
	independent	1458	

Table 1. Characteristics of the datasets used in the experimentation: number of examples (#E), number of classes (#C).

When the original dataset is divided into training and testing sets (Viral, Membrane, GRAM, Virulent) we use the testing protocol appropriate to the dataset. In the other datasets, we perform a 2-fold cross-validation test, that is repeated ten times. We then average the results.

Among the independent dataset tests, sub-sampling (e.g., 2, 5, or 10-fold cross-validation) test, and jackknife test, which are often used for examining the accuracy of a statistical prediction method (Chou and Zhang, 1995), the jackknife test was deemed the most objective that can always yield a unique result for a given benchmark dataset, as elucidated in (Chou and Shen, 2008a) and as demonstrated by Eq.50 in (Chou and Shen, 2007a). Therefore, the jackknife test has been

increasingly and widely adopted by investigators to test the power of various prediction methods (see, e.g. (Chen et al., 2009; Chou and Shen, 2007b; Chou and Shen, 2008b; Ding and Zhang, 2008; Esmaeili et al., 2010; He et al., 2010; Jiang et al., 2008; Li and Li, 2008; Lin, 2008; Lin et al., 2008; Qiu et al., 2009; Zeng et al., 2009; Zhou, 1998; Zhou et al., 2007)). However, to reduce the computational time, we adopted the 2-fold cross-validation in this study as done by many investigators with SVM as the prediction engine.

3.2 Experimental results

The first experiment is aimed at comparing the performance on the different datasets of the following configurations: *PA, NA* and *PN* (see section 2), as well as *AAG*. The methods *NA* and *PN* are proposed in this paper for the first time.

The results reported in table 2 are obtained using a fixed value for the parameter L (L=1) and λ (λ=0.5). Other internal experiments for improving the performance by testing different values of the parameters show that no significant improvement can be obtained.

DATASETS	*PA*	*NA*	*PN*
HUM	0.638	0.631	**0.701**
HEL	0.883	0.877	**0.917**
GPCR	0.930	0.916	**0.988**
GRAM	0.863	0.813	**0.905**
MEM	0.887	0.865	**0.953**
VIRAL	0.644	0.700	**0.808**
VIR1	0.700	0.705	**0.788**
VIR2	0.585	0.764	**0.859**
VIR3	0.647	0.652	**0.731**

Table 2. Performance on the different datasets for different configurations (*PA, NA, PN*) of the method *AAG*.

It is clear examining Table 2 that *PN* outperforms the other two approaches. In the following, all the experiments related to the *AAG* method are performed according to the *PN* configuration.

Now we compare the standard *NG* and *GE* with their version based on GA (named *NG ens* and *GE ens*). In order to avoid excessive computation time, the alphabets are calculated using the *NG* descriptor on the training set of the HUMAN dataset, then these alphabets are used in all the tests performed with *NG ens* and *GE ens*. The performance of *NG ens* and *NG* are very similar. *GE ens* slightly outperforms *GE*. In our opinion a better performance could be obtained if a different GA is run on each dataset to create different alphabets for the different datasets. However, there are two problems with this approach: computational time and lack of generality. Our aim in this paper is to find a generic method without a fine tuning of the parameters for each dataset.

		DATASETS								
		HUM	HEL	GPCR	GRAM	MEM	VIRAL	VIR1	VIR2	VIR3
FEATURE EXTRACTION	GE	**0.701**	0.894	0.987	0.896	0.951	0.771	0.782	0.773	0.760
	GE ens	0.695	0.917	0.989	0.901	**0.967**	**0.803**	0.781	0.735	**0.765**
	NG	0.696	0.918	0.990	**0.913**	0.958	0.743	0.789	0.777	0.754
	NG ens	0.693	**0.920**	**0.991**	0.906	0.943	0.735	**0.791**	**0.828**	0.735

Table 3. Comparison of the standard NG and GE with their version based on GA.

In Table 4 we compare the different WA approaches reported in section 3. It is clear that the combined approach DW+DL outperforms the other methods. In the following all the experiments related to the *WA* method are performed according to the *DW+DL* configuration.

		DATASETS								
		HUM	HEL	GPCR	GRAM	MEM	VIRAL	VIR1	VIR2	VIR3
FEATURE EXTRACTION	Base	0.592	0.755	0.896	0.812	0.755	0.627	0.636	0.618	0.545
	DL	0.628	0.743	0.954	0.810	0.878	**0.781**	0.601	0.508	0.531
	DW	0.678	0.871	0.991	0.916	0.946	0.731	0.798	**0.765**	0.696
	DW+ DL	**0.690**	**0.889**	**0.992**	**0.918**	**0.953**	0.755	**0.789**	0.750	**0.701**

Table 4. Comparison of the tested wavelet-based descriptors.

The experiment reported in Table 5 is aimed at comparing different solutions, both stand-alone classifiers and ensembles. Most of the methods in the comparison are implemented using the original code (shared by the original authors). The column RANK reports the average rank of the given descriptor in the tested dataset (e.g., if a descriptor always obtains the best performance in each dataset, its rank is 1). The average rank is calculated for all the methods reported in Table 5.

The following fusion approaches[9] are also reported in Table 5:

- FUS1, leave-one-out dataset where the methods are combined by sum rule;
- FUS2, all the methods, except the 2-gram (2G) and amino acid composition (AC), combined by sum rule;
- FUS3, all the datasets are used for selecting the best descriptors, then these methods are combined by sum rule. The selected descriptors of FUS3 are RC, PE, AAG, AA, and 2G.
- FUS4, leave-one-out dataset where the methods are combined by weighted sum rule.

The Sum rule selects as final score the sum of the scores of the pool of the selected approaches, the scores are the output of the SVM trained with that approach. In the weighted sum rule the scores of each approach are weighted by a weight between 0.1 and 1. Also these weights,

[9] Before the fusione the scores of each method are normalized to mean 0 and standard deviation 1

to avoid any overfitting, are selected using the same data used by SFFS for selecting the approaches to be combined.

		DATASETS									*RANK*
		HUM	HEL	GPCR	GRAM	MEM	VIRAL	VIR1	VIR2	VIR3	
FEATURE EXTRACTION	AC	0.613	0.780	0.960	0.872	0.889	0.615	0.745	0.813	0.717	15.0
	SAC	0.679	0.824	0.959	0.870	0.917	0.685	0.719	0.761	0.705	14.4
	2G	0.687	0.918	0.978	0.899	0.940	0.647	0.770	0.839	0.735	10.8
	AUC	0.704	0.901	**0.992**	**0.929**	0.926	0.754	0.762	0.824	0.744	8.6
	FS	0.667	0.786	0.981	0.857	0.880	0.660	0.722	0.805	0.728	14.4
	MK	0.665	0.768	0.988	0.702	0.936	0.718	0.754	0.790	0.735	13.2
	PE	0.696	0.919	0.956	0.866	0.947	0.651	0.780	0.845	0.721	11.2
	RC	**0.717**	***0.925***	0.991	0.880	**0.953**	0.608	**0.812**	0.872	0.730	7.3
	WA	0.690	0.889	**0.992**	0.918	**0.953**	0.755	0.789	0.750	0.701	10.1
	AAG	0.701	0.917	0.988	0.905	**0.953**	**0.808**	0.788	0.859	0.731	8.1
	AA	0.638	0.805	0.991	0.921	0.910	0.699	0.805	***0.892***	0.756	8.8
	GE	0.695	0.917	0.989	0.901	***0.967***	0.803	0.781	0.735	**0.765**	8.2
	NG	0.693	0.920	0.991	0.906	0.943	0.735	0.791	0.828	0.735	8.5
	FUS1	0.725	0.910	0.997	0.921	0.960	0.817	0.814	0.825	0.751	4.7
	FUS2	*0.732*	0.921	0.997	0.930	0.959	*0.827*	0.818	0.859	*0.769*	2.7
	FUS3	0.725	*0.925*	0.997	*0.947*	0.960	0.825	*0.829*	0.861	0.760	2.4
	FUS4	0.724	0.923	0.998	0.940	0.960	0.812	0.810	0.846	0.745	3.8

Table 5. AU obtained by different methods in the different datasets. The bold number represents the best result in a given dataset obtained by a stand-alone descriptor, the italicized numbers represent the best result in a given dataset considering all the methods.

It would be interesting to consider as baseline the simple amino acid composition (AC) and dipeptide composition (2G), since they have been widely used for protein classification.

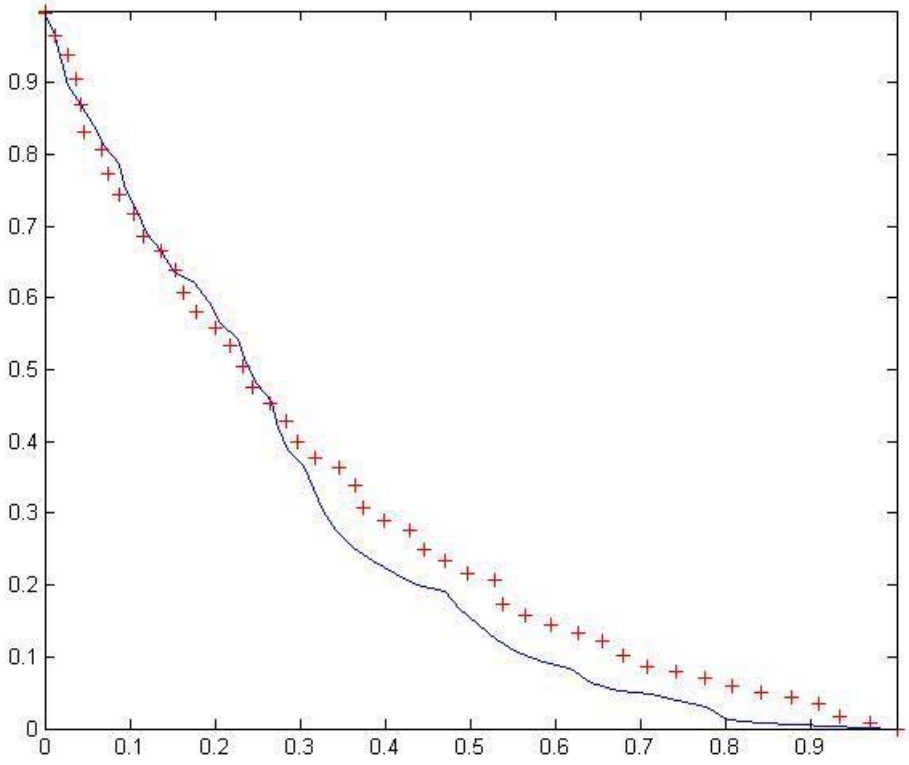
The following conclusions can be drawn from the results reported in this section:

- Our experiments, show that there is not a "best" stand-alone method that performs better than others in all the case studies; better performance stability among different test sets is obtained by combining different methods, while the performance of a single approach is influenced by the origin of the proteins evaluated;
- Combined approaches seem to be more robust, and in our experiments the ensemble named FUS4 obtains the best performance (note that FUS2 and FUS3 obtain better performance but they do not use a blind testing protocol);
- Several stand-alone approaches obtain very similar performance also if they use very different approaches for extracting features (for example, see the description of RC and AAG and PE). This is a possible motivation of the good performance of the fusion approaches. The best stand-alone method is RC, which is a modified version of the pseudo Chou's protein descriptor;
- The new descriptors proposed in this paper obtain interesting results. Our AAG variant is the second method among the stand-alone descriptors. WA does not perform as well as AAG, but we have shown that standard texture descriptors could be used for extracting features from proteins (after the wavelet transform). So in our opinion a deep study of different texture descriptor could improve the performance of WA.
- In our opinion the most interesting result is that obtained by FUS2 (it combines all the methods except AC and 2G). It obtains performance only slightly lower than that obtained by FUS3 where all the datasets are used for selecting the descriptors. It is clear the usefulness to combine a wide set of different descriptors each based on a different extraction methodology or protein representation.

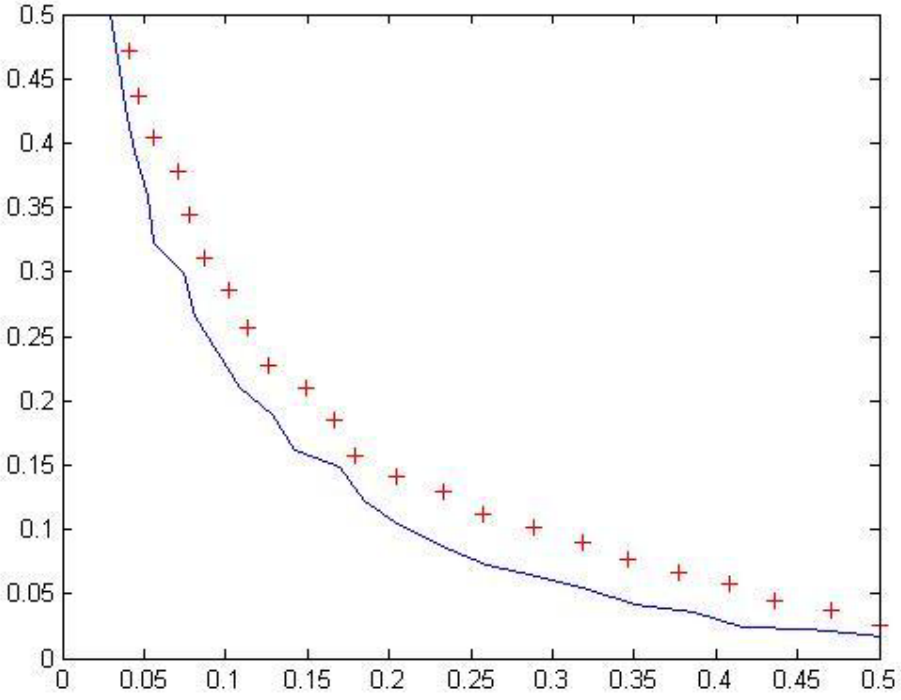
As a final experiment we report in Figure 1 the FAR/FRR-curve (false acceptation rate/false rejection rate) for the 2-class problems, obtained by AUC (the red x), the approach based on the

Chou's amino acid descriptor, the most used stand-alone feature descriptor in the literature, and our proposed fusion FUS2 (the blue line).

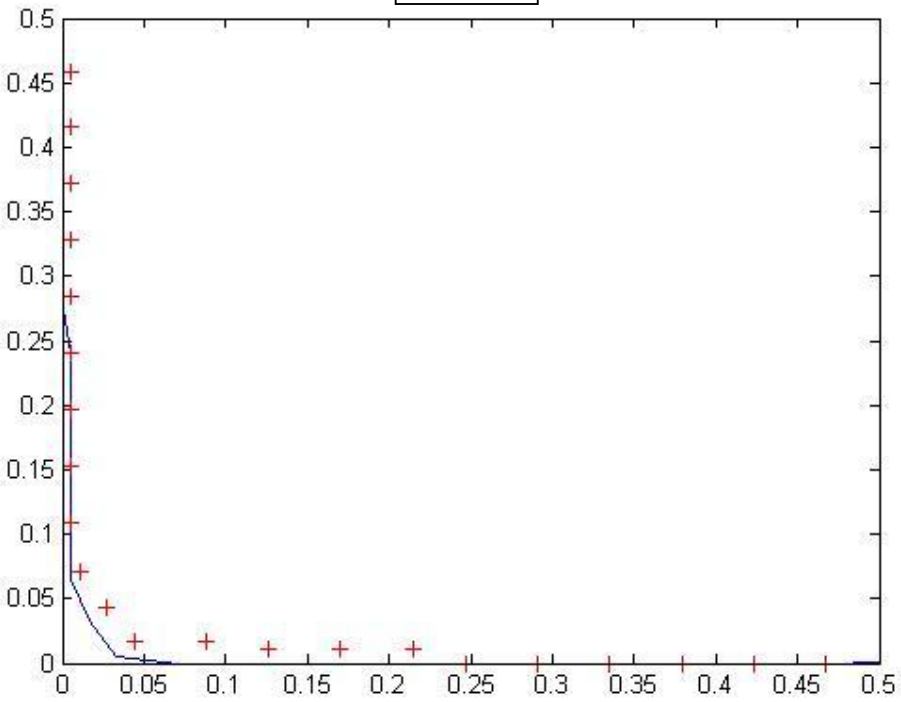
HUMAN



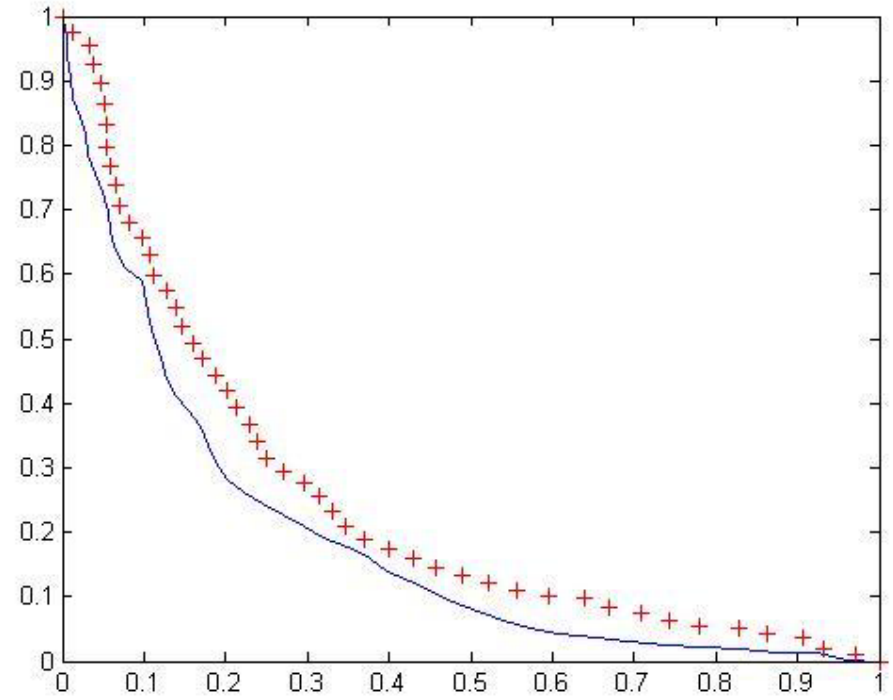
HELICO



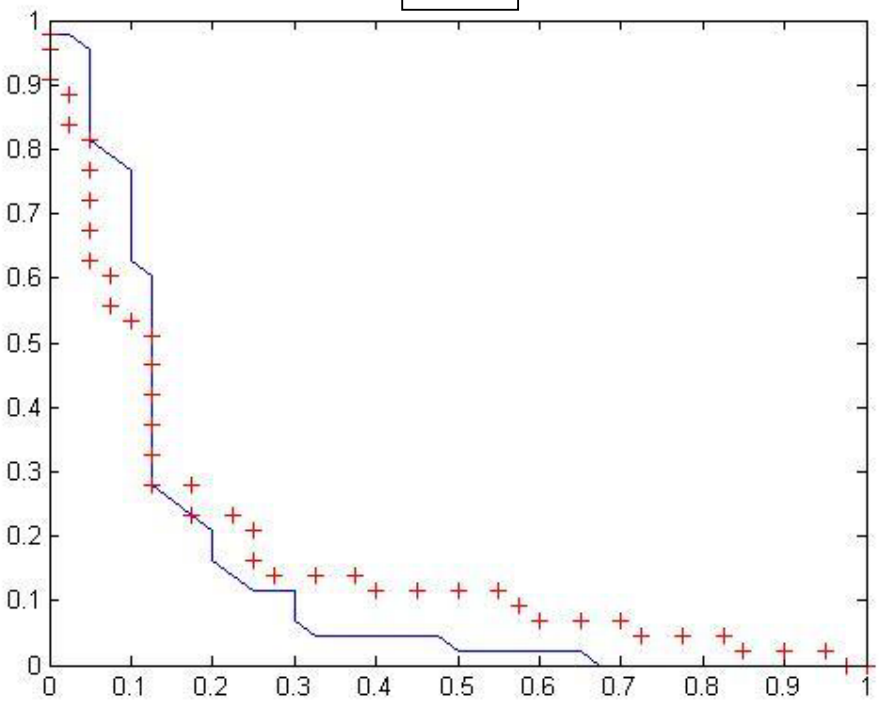
GPCR



VIR1



VIR2



VIR3

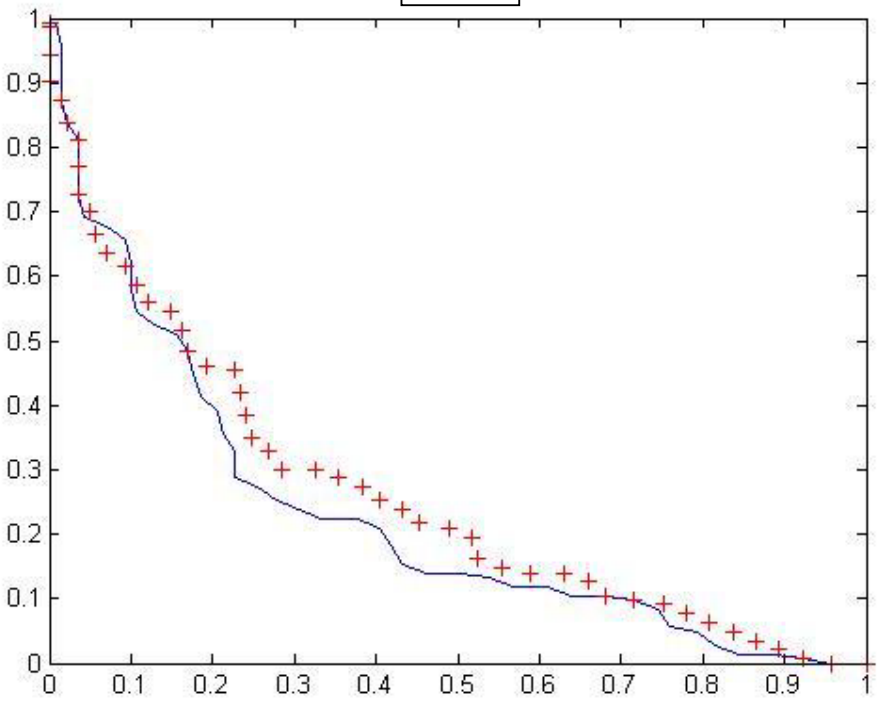


Figure 1. FAR/FRR-curve of the 2-classes problems, the y-axis is the false acceptation rate while the x-axis is the false rejection rate

These plots confirm our previous conclusions of the usefulness of the set of descriptors for classifying proteins in different classification problems.

Finally, in Table 6 we report for each tested dataset the best results reported in the literature by other authors. It is important to stress that some methods use different sources of information. For example both (Shen and Chou, 2007a) and (Shen and Chou, 2007b) are based on the fusion between ontologies information and amphiphilic pseudo amino acid composition approach, while (Garg & Gupta, 2008) and (Chou and Shen, 2007c) consider also features extracted by the position-specific scoring matrix (PSSM). In Table 6 for (Garg & Gupta, 2008) we report both results obtained considering or without considering the position-specific scoring matrix features.

Notice that the performance indicator reported in Table 6 is the accuracy (except for (Garg & Gupta, 2008) where since the scores obtained in the datasets VIR1,VIR2 and VIR3 are available it is possible to calculate the area under the ROC curve) since the cited works used as performance indicator the accuracy.

Moreover, we want to stress that several of the cited methods used as method for extracting the features from the amino-acid sequence the Chou's amino acid composition method (named in this paper AUC). From the Figure 1 it is clear that our approach outperforms AUC.

It is interesting to note that our method works well in almost all the datasets, without a parameters tuning for optimizing the performance in a given dataset. The only dataset where we obtain results far from the state-of-the-art is VIRAL, but it is important to note that (Shen and Chou, 2007a) is based on the ontologies (so the comparison is not fair).

	DATASETS								
	HUM	HEL	GPCR	GRAM	MEM	VIRAL	VIR1	VIR2	VIR3
(Garg & Gupta, 2008)	---	---	---	---	---	---	0.770	0.870	0.834
(Garg & Gupta, 2008) (No PSSM)	---	---	---	---	---	---	0.780	0.855	0.745
(Martin et al., 2005)	70.0	83.0	---	---	---	---	---	---	---
(Xiao et al., 2009a)	---	---	91.6	---	---	---	---	---	---
(Shen and Chou, 2007b)	---	---	---	84.1	---	---	---	---	---
(Shen and Chou, 2007a)	---	---	---	---	---	92.9	---	---	---
(Chou and Shen, 2007c)	---	---	---	---	92.7	---	---	---	---
FUS2	70.0	85.0	98.1	84.4	91.5	78.6	0.818	0.859	0.769

' **Table 5**. Comparison with the state-of-the-art.

4 Conclusion

In this paper, we have presented an empirical study where different feature extraction approaches for representing proteins are compared and combined. Moreover, novel configurations of the AAG method and Wavelet descriptors are proposed for the first time and evaluated. We show that we obtain the best performance when the different descriptors are combined by weighted sum rule.

We obtained a number of statistically robust observations regarding the generality and robustness of our system across an extensive evaluation of our system on different datasets. The two main conclusions that can be drawn from the results:

- Our experiments show that there is not a best stand-alone method that performs better than others on all the case studies, i.e., the best method is different for different datasets;
- Better performance stability and generality among the different test sets is obtained by combining different methods, combined approaches seem to be more robust.

To further improve the performance our method we plan on testing different classification approaches. Instead of using only SVM, we plan on investigating the performance of such ensembles of classifiers as AdaBoost and Rotation forest (Rodriguez et al., 2006). The main drawback using these ensemble methods is that they require more computational power than SVM. This is not a problem for the testing phase, but in the training phase this would be a problem if we want to compare several descriptors using several datasets.

Another way to improve performance is to use different sources of information. For example, features could be extracted directly from an analysis of the protein's spatial structure (Daras et al., 2006) or by considering the Position-Specific Scoring Matrix (PSSM) (Ben-Ga et al., 2005).

Finally, since user-friendly and publicly accessible web-servers represent the future direction for practically developing more useful predictors (Chou and Shen, 2009), we shall make efforts in our future work to provide a web-server for the method presented in this paper.

References

Bairoch, A., Apweiler, R., 2000. The SWISS-PROT protein sequence database and its supplement TrEMBL. Nucleic Acids Research 28, 45-8.

Ben-Ga, l. I., Shani, A., Gohr, A., Grau, J., Arviv, S., Shmilovici, A., Posch, S., Grosse, I., 2005. Identification of transcription factor binding sites with variable-order bayesian networks. Bioinformatics 21, 2657-2666.

Bock, J., Gough, D., 2003. Whole-proteome interaction mining. Bioinformatics 19, 125-135.

Chen, C., Chen, L., Zou, X., Cai, P., 2009. Prediction of protein secondary structure content by using the concept of Chou's pseudo amino acid composition and support vector machine. Protein & Peptide Letters 16, 27-31.

Chen, L., Yang, J., Yu, J., Yao, Z., Sun, L., Shen, Y., Jin, Q., 2005. VFDB: a reference database for bacterial virulence factors. Nucleic Acids Research 33, D325-D328.

Chou, K. C., 2001. Prediction of protein cellular attributes using pseudo-amino acid composition. PROTEINS: Structure, Fucntion, and Genetics 43, 246-255.

Chou, K. C., 2009. Pseudo amino acid composition and its applications in bioinformatics, proteomics and system biology. Current Proteomics 6, 262-274.

Chou, K. C., Zhang, C. T., 1995. Review: Prediction of protein structural classes. Critical Reviews in Biochemistry and Molecular Biology 30, 275-349.

Chou, K. C., Shen, H. B., 2007a. Review: Recent progresses in protein subcellular location prediction. Analytical Biochemistry 370.

Chou, K. C., Shen, H. B., 2007b. Signal-CF: a subsite-coupled and window-fusing approach for predicting signal peptides. Biochemical and Biophysical Research Communications 357, 633-640.

Chou, K. C., Shen, H. B., 2007c. MemType-2L: A Web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM. Biochem Biophys Res Comm 360, 339-345.

Chou, K. C., Shen, H. B., 2008a. Cell-PLoc: A package of web-servers for predicting subcellular localization of proteins in various organisms. Nature Protocols 3, 153-162.

Chou, K. C., Shen, H. B., 2008b. ProtIdent: A web server for identifying proteases and their types by fusing functional domain and sequential evolution information. Biochemical and Biophysical Research Communications 376, 321-325.

Chou, K. C., Shen, H. B., 2009. Review: Recent advances in developing web-servers for predicting protein attributes. Natural Science 2, 63-92 (openly accessible at http://www.scirp.org/journal/NS/).

Chou, K. C., Shen, H. B., 2010. A new method for predicting the subcellular localization of eukaryotic proteins with both single and multiple sites: Euk-mP Loc 2.0. PLoS ONE 5 e9931.

Cristianini, N., Shawe-Taylor, J., 2000. An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge, UK.

Daras, P., Zarpalas, D., Axenopoulos, A., Tzovaras, D., Strintzis, M. G., 2006. Three-dimensional shape-structure comparison method for protein classification. IEEE Transactions on Computational Biology and Bioinformatics 3, 193-207.

Ding, Y. S., Zhang, T. L., 2008. Using Chou's pseudo amino acid composition to predict subcellular localization of apoptosis proteins: an approach with immune genetic algorithm-based ensemble classifier. Pattern Recognition Letters 29, 1887-1892.

Esmaeili, M., Mohabatkar, H., Mohsenzadeh, S., 2010. Using the concept of Chou's pseudo amino acid composition for risk type prediction of human papillomaviruses. Journal of Theoretical Biology 263, 203-209.

Fawcett, T., 2004. ROC graphs: Notes and practical considerations for researchers. HP Laboratories, Palo Alto, USA.

Garg, A., Gupta, D., 2008. VirulentPred: a SVM based prediction method for virulent proteins in bacterial pathogens. BMC Bioinformatics 9, doi:10.1186/1471-2105-9-62.

Guo, J., Lin, Y., Sun, Z., 2005. A novel method for protein subcellular localization: Combining residue-couple model and SVM. Proceedings of 3rd Asia-Pacific Bioinformatics Conference, pp. 117-129.

He, Z. S., Zhang, J., Shi, X. H., Hu, L. L., Kong, X. G., Cai, Y. D., Chou, K. C., 2010. Predicting drug-target interaction networks based on functional groups and biological features. PLoS ONE 5, e9603.

Ho, T. K., 1998. The random subspace method for constructing decision forests. IEEE PAMI 20, 832-844.

Hu, J., Zhang F., Improving Protein Localization Prediction Using Amino Acid Group Based Physichemical Encoding, BICoB 2009, LNBI 5462, pp. 248–258, 2009.

Jaakkola, T., Diekhans, M., Haussler, D., 1999. Using the fisher kernel method to detect remote protein homologies. Seventh International Conference on Intelligent Systems for Molecular Biology. AAAI Press, pp. 149-158.

Jiang, X., Wei, R., Zhang, T. L., Gu, Q., 2008. Using the concept of Chou's pseudo amino acid composition to predict apoptosis proteins subcellular location: an approach by approximate entropy. Protein & Peptide Letters 15, 392-396.

Kawashima, S., Kanehisa, M., 2000. AAindex: amino acid index database. Nucleic Acids Research 20.

Kittler, J., 1998. On combining classifiers. IEEE PAMI 20, 226–239.

Kumar, M., Verma, R., G. P. S, R., 2005. Prediction of mitochondrial proteins using support vector machine and hidden markov model. Journal of Biological Chemistry 281, 5357-5363.

Landgrebe, T.C.W., Duin, Robert P.W., 2007. Approximating the multiclass ROC by pairwise analysis. Pattern Recognition Letters 28 (2007) 1747–1758

Lei, Z., Dai, Y., 2005. An SVM-based system for predicting protein subnuclear localizations. BMC Bioinformatics 6.

Leslie, C. S., Eskin, E., Cohen, A., Weston, J., Noble, W. S., 2004. Mismatch string kernels for discriminative protein classification. Bioinformatics 20, 467-476.

Li, F. M., Li, Q. Z., 2008. Predicting protein subcellular location using Chou's pseudo amino acid composition and improved hybrid approach. Protein & Peptide Letters 15.

Li, Z.-C., Zhou, X.-B., Dai, Z., Zou, X.-Y., 2008. Prediction of protein structural classes by Chou's pseudo amino acid composition: approached using continuous wavelet transform and principal component analysis Amino Acids DOI 10.1007/s00726-008-0170-2 22, 699-705.

Liao, S., Law, M. W. K., Chung, A. C. S., 2009. Dominant local binary patterns for texture classification. IEEE Transactions on Image Processing 18, 1107 – 1118.

Lin, H., 2008. The modified Mahalanobis discriminant for predicting outer membrane proteins by using Chou's pseudo amino acid composition. Journal of Theoretical Biology 252, 350-356.

Lin, H., Ding, H., Feng-Biao Guo, F. B., Zhang, A. Y., Huang, J., 2008. Predicting subcellular localization of mycobacterial proteins by using Chou's pseudo amino acid composition. Protein & Peptide Letters 15, 739-744.

Lin, W. Z., Xiao, X., Chou, K. C., 2009. GPCR-GIA: A web-server for identifying G-protein coupled receptors and their families with grey incidence analysis. Protein Engineering, Design & Selection

Martin, S., Roe, D., Faulon, J.L., 2005. Predicting protein-protein interactions using signature products. Bioinformatics, 21(2):218-226

Murphy, L. R., Wallqvist, A., Levy, R. M., 2000. Simplified amino acid alphabets for protein fold recognition and implications for folding. Protein Engineering 13, 149-52.

Nanni, L., 2006. Comparison among feature extraction methods for HIV-1 Protease Cleavage Site Prediction, Pattern Recognition, 39, 711–713

Nanni, L., Lumini, A., 2006. An ensemble of K-Local Hyperplane for predicting Protein-Protein interactions. BioInformatics 22, 1207-1210.

Nanni, L., Lumini, A., 2008. A genetic approach for building different alphabets for peptide and protein classification. BMC Bioinformatics 9.

Pudil, P., Novovicova, J., Kittler, J., 1994. Floating search methods in feature selection. Pattern Recognition Letters 5, 1119-1125.

Qin, Z. C., 2006. ROC analysis for predictions made by probabilistic classifiers. Fourth International Conference on Machine Learning and Cybernetics, Vol. 5, pp. 3119-312.

Qiu, J. D., Huang, J. H., Liang, R. P., Lu, X. Q., 2009. Prediction of G-protein-coupled receptor classes based on the concept of Chou's pseudo amino acid composition: an approach from discrete wavelet transform. Analytical Biochemistry 390, 68-73.

Rodriguez, J. J., Kuncheva, L. I., Alonso, C. J., 2006. Rotation forest: a new classifier ensemble method. IEEE PAMI 28, 1619-1630.

Shen, H.-B., Chou, K.-C., 2007a. Virus-PLoc: A fusion classifier for predicting the subcellular localization of viral proteins within host and virus-infected cells. Biopolymers 15, 233-240.

Shen, H.-B., Chou, K.-C., 2007b. Gpos-PLoc: an ensemble classifier for predicting subcellular localization of Gram-positive bacterial proteins. Protein Engineering Design & Selection 20, 39-46.

Shen, H. B., Chou, K. C., 2010. Gneg-mPLoc: A top-down strategy to enhance the quality of predicting subcellular localization of Gram-negative bacterial proteins. Journal of Theoretical Biology 264, 326-333.

Tan, X., Triggs, B., 2007. Enhanced local texture feature sets for face recognition under difficult lighting conditions. Analysis and Modelling of Faces and Gestures LNCS 4778, 168-182.

Tantoso, E., Li, K. B., 2008. AAIndexLoc: predicting subcellular localization of proteins based on a new representation of sequences using amino acid indices. Amino Acids 35, 345-53.

Verma, R., Varshney, G. C., Raghava, G. P. S., 2009. Prediction of mitochondrial proteins of malaria parasite using split amino acid composition and PSSM profile. Amino Acids DOI 10.1007/s00726-009-0381-1.

Xi, L., Bo, L., Yu, S., Qingguang, Z., Jiawei, L., 2009. Protein functional class prediction using global encoding of amino acid sequence Protein functional class prediction using global encoding of amino acid sequence. Journal of Theoretical Biology 261, 290-293.

Xiao, X., Lin, W. Z., 2009. Application of protein grey incidence degree measure to predict protein quaternary structural types. Amino Acids 37, 741-749.

Xiao, X., Wang, P., Chou, K. C., 2008a. Predicting protein structural classes with pseudo amino acid composition: an approach using geometric moments of cellular automaton image. Journal of Theoretical Biology 254, 691-696.

Xiao, X., Lin, W. Z., Chou, K. C., 2008b. Using grey dynamic modeling and pseudo amino acid composition to predict protein structural classes. Journal of Computational Chemestry 29, 2018-2024.

Xiao, X., Wang, P., Chou, K. C., 2009a. GPCR-CA: A cellular automaton image approach for predicting G-protein-coupled receptor functional classes. Journal of Computational Chemistry 30, 1414-1423.

Xiao, X., Wang, P., Chou, K. C., 2009b. Predicting protein quaternary structural attribute by hybridizing functional domain composition and pseudo amino acid composition. Journal of Applied Crystallography 42, 169-173.

Xiao, X., Shao, S. H., Huang, Z. D., Chou, K. C., 2006a. Using pseudo amino acid composition to predict protein structural classes: approached with complexity measure factor. Journal of Computational Chemestry.

Xiao, X., Shao, S. H., Ding, Y. S., Huang, Z. D., Chou, K. C., 2006b. Using cellular automata images and pseudo amino acid composition to predict protein subcellular location. Amino Acids 30, 49-54.

Xiao, X., Shao, S., Ding, Y., Huang, Z., Huang, Y., Chou, K. C., 2005. Using complexity measure factor to predict protein subcellular location. Amino Acids 28, 57-61.

Yang, Z. R., Thomson, R., 2005. Bio-basis function neural network for prediction of protease cleavage sites in proteins. IEEE Transactions on Neural Networks 16, 263-274.

Zeng, Y. H., Guo, Y. Z., Xiao, R. Q., Yang, L., Yu, L. Z., Li, M. L., 2009. Using the augmented Chou's pseudo amino acid composition for predicting protein submitochondria locations based on auto covariance approach. Journal of Theoretical Biology 259, 366-72.

Zhou, G. P., 1998. An intriguing controversy over protein structural class prediction. Journal of Protein Chemestry 17, 729-738.

Zhou, X. B., Chen, C., Li, Z. C., Zou, X. Y., 2007. Using Chou's amphiphilic pseudo-amino acid composition and support vector machine for prediction of enzyme subfamily classes. Journal of Theoretical Biology 248, 546-551.