Amino Acids (2013) 44:887–901
DOI 10.1007/s00726-012-1416-6

ORIGINAL ARTICLE

An empirical study on the matrix-based protein representations and their combination with sequence-based approaches

Loris Nanni · Alessandra Lumini · Sheryl Brahnam

Received: 15 March 2012 / Accepted: 3 October 2012 / Published online: 30 October 2012

Abstract Many domains have a stake in the development of reliable systems for automatic protein classification. Of particular interest in recent studies of automatic protein classification is the exploration of new methods for extracting features from a protein that enhance classification for specific problems. These methods have proven very useful in one or two domains, but they have failed to generalize well across several domains (i.e. classification problems). In this paper, we evaluate several feature extraction approaches for representing proteins with the aim of sequence-based protein classification. Several protein representations are evaluated, those starting from: the position specific scoring matrix (PSSM) of the proteins; the amino-acid sequence; a matrix representation of the protein, of dimension (length of the protein) $\times 20$, obtained using the substitution matrices for representing each amino-acid as a vector. A valuable result is that a texture descriptor can be extracted from the PSSM protein representation which improves the performance of standard descriptors based on the PSSM representation. Experimentally, we develop our systems by comparing several protein descriptors on nine different datasets. Each descriptor is used to train a support vector machine (SVM) or an ensemble of SVM. Although different stand-alone descriptors work well on some datasets (but not on others), we have discovered that fusion among classifiers trained using different descriptors obtains a good performance across all the tested datasets. Matlab code/Datasets used in the proposed paper are available at http://www.bias.csr.unibo.it\nanni\PSSM.rar.

Keywords Proteins classification · Machine learning · Ensemble of classifiers · Support vector machines

L. Nanni (✉)
DEI, University of Padua, viale Gradenigo 6, Padua, Italy
e-mail: loris.nanni@unibo.it

A. Lumini
DEIS, Università di Bologna, Via Venezia 52,
47521 Cesena, Italy
e-mail: alessandra.lumini@unibo.it

S. Brahnam
Computer Information Systems, Missouri State University,
901 S. National, Springfield, MO 65804, USA
e-mail: sbrahnam@missouristate.edu

Introduction

Until fairly recently, classification of proteins has been largely a manual process. Efforts to automate the process have become a major area of study in machine learning (Wang et al. 1994). Central to this work is discovering good methods of extracting features from proteins for enhancing the classification process for different applications (Maddouri and Elloumi 2004, Saidi et al. 2010), such as subcellular localization and protein–protein interactions (Chou and Shen 2007b, 2010, Shen and Chou 2010).

Protein feature extraction methods fall into three main classes. The first class considers Chou's pseudo amino acid (PseAA) composition (Chou and Shen 2007b). PseAA has proven to be one of the most studied and most popular methods for extracting features from proteins. It expands the simple amino acid composition (AAC) by retaining information embedded in protein sequences (Chou 2001, 2009). The PseAA model does this by including some additional factors that incorporate information regarding a protein's sequential order along with the first 20 factors representing the components of the conventional AAC. Various modes, e.g., a series of rank-different correlation factors along a protein chain, are utilized to represent the

sequential information. An excellent history of the development of PseAA that includes how to use the concept of Chou's PseAAC to develop 16 variant forms of PseAAC can be found in (Chou 2009). In this paper, we will consider additional forms of PseAAC and show how they further strengthen the power of PseAAC.

The second class of protein feature extraction methods includes techniques that are based on a vectorial representation of the protein. In these methods, feature extraction is not explicitly related to groups of amino acids and the protein is represented by a vector of fixed length. Given a protein sequence $P = (p_1, p_2, \ldots, p_N)$ where $p_i \in \mathcal{A} = [A, C, D, \ldots, Y]$, P is represented by a vector $\in \Re^N$ without explicitly considering how the amino acids are grouped inside a given protein.

In (Nanni et al. 2010), for instance, some physicochemical encodings are reported that combine the value of a given property for an amino acid with its 2-grams representation. Another vectorial representation is the quasi residue couple (Nanni 2006), a model which combines the information related to a fixed physicochemical property of the protein with the sequence order effect of the composition of the amino acid.

The third class of protein feature extraction methods includes techniques that are based on kernels. The Fisher kernel (Jaakkola et al. 1999) was one of the first proposed for remote homology detection. A kernel that performs similarly but with less computational cost is the mismatch string kernel proposed in (Leslie et al. 2002, 2004). It measures sequence similarity based on shared occurrences of subsequences. Yet, another class of kernels was developed by (Lei and Dai 2005) for vectors derived from *k*-peptide vector, mapped by a matrix of high-scored pairs of k-peptides measured by BLOSUM62 scores. These kernel functions were used for training a support vector machine (SVM). In (Yang and Thomson 2005), the biobasis function neural network was trained with sequence distances obtained using sequence alignment.

Different pseudo amino acid compositions have been developed and widely used for very specific practical applications. Some biological sequence feature representations designed for predicting various biological attributes include cellular automata image classification (Lin et al. 2009; Xiao et al. 2006b, 2008b, 2009a), complexity measure factor (Xiao et al. 2005, 2006a); grey dynamic model (Xiao and Lin 2009, Xiao et al. 2008a); and functional domain composition (Xiao et al. 2009b). Despite the promise held by machine learning, sequence-based function prediction remains a challenge. Critical issues center on a lack of understanding regarding the biological properties that characterize protein function, thereby making it difficult to extract informative features. This paper focuses on sequence-based protein classification; we evaluate novel features for protein function prediction. Some of these features are derived from the position-specific scoring matrix (Gribskov et al. 1987), which describes a protein starting from the evolutionary information contained in a PSI-BLAST similarity search.

The main objective of this study is to search for a general ensemble method that could works well on different protein classification datasets and problems. Studying protein classification methods that generalize well has the potential of deepening our understanding of protein representation and of promoting the development of more robust and powerful classification systems. Such investigations also have the potential of speeding up real world development in new areas involving protein classification.

To reach our goal, we extract some descriptors from a matrix representation of amino acids, which we call the substitution matrix representation (SMR). This representation is obtained by replacing each amino-acid of the sequence with a row in a given substitution matrix. We propose new protein descriptors for matrix representation that can be extracted from both the PSSM matrix and the SMR matrices. A valuable result is that a well-known texture descriptor named local phase quantization can be coupled with a standard approach based on PSSM representation for improving its performance (see the method named WS2 in the "Experimental" section).

To obtain a system that generalizes well, we perform an exhaustive search for the best ensemble based on the combination of our novel representations along with the best known descriptors for peptides/proteins.

The remainder of this paper is organized as follows. In "Pattern representation and feature extraction", we introduce the feature extraction methods explored in this work. In "Experiments", we report experimental results obtained using datasets for different classification problems. Finally, in "Conclusions", we draw a number of conclusions.

Pattern representation and feature extraction

Recent research has focused on finding a compact and effective representation of proteins because there are many classification problems (e.g., subcellular localization and protein–protein interactions) that require a machine learning approach (Chou and Shen 2007a; Nanni et al. 2010). One feasible solution is base extraction on a fixed length encoding scheme and couple it with a general purpose classifier.

In this work, we test three different representations for proteins: two based on a matrix representation and a third based on the classical amino acid primary sequence. From each of these representations, a set of fixed length

descriptors is extracted. Below, we briefly describe the representations explored in this study.

We are able to use SVM[1] as the classifier, because we adopted a fixed length encoding scheme. SVM is arguably one of the most widely used techniques for classification. It arose from the field of statistical learning theory (Cristianini and Shawe-Taylor 2000) and is a binary-class prediction method. The basic idea behind SVM is to find the equation of a hyperplane that divides the training set into two classes, with all the points of the same class on the same side, while simultaneously maximizing the distance between the two classes and the hyperplane. In those cases where a linear decision boundary does not exist, kernel functions are used to project the data onto a higher-dimensional feature space so that they can be separated by a hyperplane. Some typical kernels include polynomial kernels and the radial basis function kernel. In our experiments, all features used for training a SVM are linearly normalized to [0 1] considering the training data. In each dataset, the SVM is tuned considering only the training data (the test is blind) using a gird search approach (Cristianini and Shawe-Taylor 2000).

When a set of descriptors is extracted instead of a single one, for each descriptor a different SVM is trained, and the final decision is obtained by combining the pool of SVMs by the sum rule. The sum rule simply sums the scores obtained by the pool of SVMs classifiers: let us define SV_i the set of scores obtained by the i-th SVM and k the number of combined SVMs, then the scores of the ensemble is given by: $\sum_{i=1,\ldots,k} SV_i$.

The rest of this section includes the explanation of two descriptors extracted from the classical amino acid primary sequence (Quasi Residue Couple in "Quasi Residue Couple (RC)" and Autocovariance approach in sub-Sect. "Autocovariance approach (AC)") and several descriptors (sub-Sects. "Matrix-based descriptors" and "Texture descriptors") obtained by a matrix representation of the protein (Position Specific Scoring Matrix in Sub-sect. "A matrix representation for proteins: position Specific Scoring Matrix" and Substitution matrix representation in Sub-sect. "A new matrix representation for proteins: substitution matrix representation"). A summary of all the descriptors is reported in Table 2.

A primary representation for proteins: amino acid sequence (AAS)

The older and most used representation for proteins is the simple amino acid sequence, which is a linear sequence of amino acids. The amino acid representation for a given protein sequence is

$P = (p_1, p_2, \ldots, p_N)$

where $p_i \in \mathcal{A} = [A, C, D, \ldots, Y]$. This representation has been proven to be outperformed by several newer representation methods, mainly based on the use of physicochemical properties of amino acids (Kawashima and Kanehisa 1999). In this work, we use two descriptors based on the combination of AAS with physicochemical properties of amino acids: Quasi Residue Couple (Nanni et al. 2010) and Autocovariance approach (Zeng et al. 2009).

Quasi residue couple (RC)[2]

Quasi Residue Couple is a model for protein representation that was inspired by Chou's quasi-sequence-order model and Yuan's Markov chain model (Guo et al. 2005). The RC encoding scheme combines 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 is considered to represent the sequence. For each nonzero entry in the residue couple, a corresponding value of the selected property is selected.

The RC model (order $m \leq 3$) for a physicochemical property d, is given by:

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

where i and j are the 20 different amino acids; N is the length of the protein; d is 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 amino acid in location b is j. $H_{i,j}(a, b, d) = 0$ otherwise.

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

We obtain the set of physicochemical properties using the amino acid index (Kawashima and Kanehisa 1999) database.[3] This database currently contains 544 indices and 94 substitution matrices. The amino acid index is a set of 20 numerical values, with each value representing the different physicochemical properties of amino acids. One

[1] SVM is implemented as in the LibSVM toolbox http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[2] Matlab code: http://bias.csr.unibo.it/nanni/QRcouple2.zip.

[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.

problem using this database is that many physicochemical properties are highly correlated with each other. We randomly select 50 physicochemical properties, as motivated in (Nanni et al. 2010).

Autocovariance approach (AC)[4]

The Autocovariance approach (Zeng et al. 2009) is a sequence-based variant of the Chou's pseudo amino acid composition. A set of pseudo-amino-acid-based features[5] are extracted from a given protein as the concatenation of the 20 standard amino acid composition values and m values reflecting the effect of sequence order (where m is a parameter, here $m = 20$, denoting the maximum distance between two considered amino acids i, j). The final descriptor is a vector $C = (C_1, \ldots, C_{20}, C^d_{20+1}, \ldots, C^d_{20+m})$ where $C_1, \ldots, C_{20}$ are the 20 standard amino acid composition values and $C^d_{20+1}, \ldots, C^d_{20+m}$ are defined as:

$$C^d_{20+l} = \sum_{\substack{k=1 \\ l \in [1 \ldots m]}}^{N-l} \frac{(\text{index}(A(k),d) - M_\text{d}) \cdot (\text{index}(A(k+l),d) - M_\text{d})}{V_\text{d} \cdot (\text{Len} - l)}$$

where $A(k)$ denotes the index of the amino acid in the kth position of the protein; N 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 M_d and V_d are normalization factors denoting the average and the variance of the physicochemical property d on the 20 amino acids.

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

$$V_\text{d} = \frac{1}{20} \sum_{i=1}^{20} (\text{index}(i,d) - M_\text{d})^2$$

In this work, we consider the vector $C = (C_1, \ldots, C_{20}, C^d_{20+1}, \ldots, C^d_{20+m})$ calculated using 50 physicochemical properties (thus obtaining 50 descriptors).

A matrix representation for proteins: position specific scoring matrix (PSSM)

PSSM,[6] first introduced in (Gribskov et al. 1987) for detecting distantly related proteins, is generated from a group of sequences previously aligned by structural or sequence similarity. Position-specific iterated BLAST (PSI-BLAST) is the most commonly used application. It compares PSSM profiles for detecting remotely related homologous proteins or DNA.

PSSM considers the following parameters:

1. Position, which indicates the sequentially increased index of each amino acid residue in a sequence after multiple sequence alignment;
2. Probe, which is a group of typical sequences of functionally related proteins that have been aligned by sequence or structural similarity;
3. Profile, which is a matrix of 20 columns corresponding to the 20 amino acids;
4. Consensus, which is the sequence of amino acid residues that is most similar to all the alignment residues of probes at each position. The consensus sequence is generated by selecting the highest score in the profile at each position.

A PSSM representation for a given protein sequence of length N is an $N \times 20$ matrix. Each element PSSM(i,j) of the matrix is obtained by the following formula:

$$\text{PSSM}(i,j) = \sum_{k=1}^{20} w(i,k) \times Y(j,k) \quad i = 1..N, j = 1..20$$

where $w(i,k)$ is the ratio between the frequency of appearing the kth amino acid (among the 20 amino acids) at the position i of the probe and total number of probes, and $Y(j,k)$ is the value of Dayhoff's mutation matrix between the jth and kth amino acids ($Y(j,k)$ is a "substitution matrix"[7]) .

A large value of the score PSSM(i,j) indicates a highly conserved position and a small value indicating a weakly conserved position. While the formations of PSSM are slightly different from one application to another, the principles are very much the same. In our study, we used PSI-BLAST to create PSSMs for each protein sequence.

A new matrix representation for proteins: substitution matrix representation

We named Substitution Matrix Representation (SMR), a variant of a representation method proposed by (Yu et al. 2011). A SMR for a given protein of length N is a $N \times 20$ matrix where each element SMR(i,j) of the matrix is obtained as[8]:

[4] Matlab code: http://bias.csr.unibo.it/nanni/EstraggoFeaturesAC.rar.

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

[6] For extracting PSSM, after installation of PSI-BLAST, with Matlab to use system('blastpgp.exe -i input.txt -d D:\PSI-BLAST\swissprot -Q PSSM.txt -j 3'); where: input.txt is the protein sequence; PSSM.txt contains the PSSM matrix.

[7] A "substitution matrix" describes the rate at which one character in a protein sequence changes to other character states over time. Substitution matrices are usually seen in the context of amino acid or DNA sequence alignments, where the similarity between sequences depends on their divergence time and the substitution rates as represented in the matrix. (http://en.wikipedia.org/wiki/Substitution_matrix, Accessed 07/14/2012).

[8] Matlab code: http://bias.csr.unibo.it/nanni/SMR.rar.

$$\mathrm{SMR}(i,j) = M(P(i),j) \quad i = 1\ldots N, j = 1\ldots 20$$

where M is a 20 × 20 substitution matrix, whose element $M_{i,j}$ represents the probability of amino acid i mutating to amino acid j during the evolution process and $P = (p_1, p_2, \ldots, p_N)$ is the given protein.

The representation matrix SMR of a given protein has the same dimension of its PSSM, therefore they can be used to calculate the same set of fixed-length descriptors. Here, we have used the "STROMA score matrix for the alignment of known distant homologs" (Kawashima and Kanehisa 1999) as substitution matrix since it works well on average in the tested datasets, to improve the performance it is possible to use different substitution matrix for building an ensemble of SMRs.

Matrix-based descriptors

In this subsection, the fixed length descriptors extracted both from PSSM and SMR matrix representations are explained.

Average blocks

This descriptor, originally proposed by (Jeong et al. 2011) for the PSSM representation, is based on a local average of the input matrix Mat of dimension $N \times 20$. The descriptor AB is a fixed length vector of 400 elements:

$$\mathrm{AB}(k) = \frac{20}{N}\sum_{z=1}^{N/20} \mathrm{Mat}\left(z + (i-1) * \frac{N}{20}, j\right)$$
$$i = 1\ldots 20, j = 1\ldots 20,\ k = j + 20 \times (i-1)$$

where k is a linear index used to scan the cells of Mat. Thus, the final descriptor is a vector obtained as the average of Mat blocks (each related to the 5 % of a sequence).

Single average

This descriptor (Garg and Gupta 2008) is a variant of the previous one, designed to consider domains of a sequence with similar conservation rates. The rationale of this descriptor is to group together rows related to the same amino acid.

We tested two variant of the single average descriptor:SA performs a matrix normalization using a sigmoid function by which each matrix element is scaled to a range [0, 1], SA1[9] does not perform normalization.

The descriptors SA (SA1) is a fixed length vector of 400 elements:

$$\mathrm{SA}(k) = \mathrm{avg}_{i=1..N}\mathrm{Mat}(i,j) * \delta(P(i), A(z))$$
$$z = 1..20, j = 1..20, k = j + 20 \times (z-1)$$

where k is a linear index used to scan the cells of Mat, and $\delta(\cdot,\cdot)$ is the delta function.

Average similar cells

This descriptor (Jeong et al. 2011) is a variant of the previous one, which considers only positive cells as the average. The descriptor ASC is a fixed length vector of 400 elements:

$$\mathrm{ASC}(k) = \mathrm{avg}_{i=1..N}\rho(\mathrm{Mat}(i,j)) * \delta(P(i), \mathcal{A}(z))$$
$$z = 1,\ldots,20, j = 1,\ldots,20 \times (z-1)$$

where k is a linear index used to scan the cells of Mat, $\mathcal{A} = [\mathrm{A,C,D,\ ...,Y}]$ is the ordered set of amino acids, P is the given protein, $\delta(\cdot,\cdot)$ is the delta function and $\rho(\cdot)$ is a function that selects only positive values (therefore only positive cells are considered in the average).

$$\rho(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Autocovariance matrix

This descriptor proposed in (Yang et al. 2010) is aimed at avoiding the loss of the local sequence-order information. Autocovariance variables are applied to each column of the input matrix to reduce each column to a fixed length of one. In this work, an autocovariance matrix (AM) is used to describe the average correlation between positions, with a series of lag (i.e., the residue number when applied to protein sequences) apart throughout the protein sequence. AM can be calculated as follows:

$$\mathrm{AM}(k) = \frac{1}{N-\mathrm{lag}}\sum_{i=1}^{N-\mathrm{lag}}\left(\mathrm{Mat}(i,j) - \frac{1}{N}\sum_{i=1}^{N}\mathrm{Mat}(i,j)\right) \times \left(\mathrm{Mat}(i+\mathrm{lag},j) - \frac{1}{N}\sum_{i=1}^{N}\mathrm{Mat}(i,j)\right)$$
$$j = 1..20, \mathrm{lag} = 1..15, k = j + 20 \times (\mathrm{lag}-1)$$

where k is a linear index used to scan the cells of Mat, lag denotes the distance between one residue and its neighbors and N is the length of the sequence.

Pseudo PSSM

This Pseudo PSSM approach (PP)[10] is one of the most used descriptor (e.g. Fan and Li 2011; Jeong et al. 2011) for protein giving its PSSM. In this work, we extend its use also to SMR. To avoid complete loss of the sequence-order information, the concept of the pseudo amino acid composition is adopted. Given an input matrix *Mat* of dimension $N \times 20$, we define:

[9] Matlab code: http://bias.csr.unibo.it/nanni/SA1.rar.

[10] Matlab code: http://bias.csr.unibo.it/nanni/PP.rar.

$$E(i,j) = \frac{\mathrm{Mat}(i,j) - \frac{1}{20}\sum_{v=1}^{20}\mathrm{Mat}(i,v)}{\sqrt{\frac{1}{20}\sum_{u=1}^{20}\left(\mathrm{Mat}(i,u) - \frac{1}{20}\sum_{v=1}^{20}\mathrm{Mat}(i,v)\right)^2}} \quad i = 1\ldots20, j = 1\ldots20$$

$$\mathrm{LTP}(P,R) = \sum_{p=0}^{P-1} s(u_p - x)2^p \quad \text{where} \quad s(x) = \begin{cases} 1, x > \tau \\ -1, x < -\tau \\ 0, |x| \le \tau \end{cases}$$

$$\mathrm{PP}(k) = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} E(i,j) & k = 1,\ldots,20 \\ \frac{1}{N-\mathrm{lag}}\sum_{i=1}^{N-\mathrm{lag}}[E(i,j) - E(i+\mathrm{lag},j)]^2 & j = 1,\ldots,20, \mathrm{lag} = 1\ldots15 \\ & k = 20 + j + 20\cdot(\mathrm{lag}-1) \end{cases}$$

The final descriptor is a vector PP of length 320.

Texture descriptors

Both PSSM and SMR matrix representations for proteins can be treated as images and used to extract texture descriptors. In this work, we test two high performing descriptors (detailed in the following sub-sections): local ternary pattern (LTP) (Tan and Triggs 2007) and local phase quantization (LPQ)[11] (Ojansivu and Heikkila 2008). Both the descriptors are extracted according to a global and a local evaluation after that the matrix/image has been normalized between 0 and 255 (as a standard image).

According to the different methods, we obtain the following couple of descriptors:

$\mathrm{ImG}_{\mathrm{LTP}}$ and $\mathrm{ImG}_{\mathrm{LPQ}}$ are obtained by extracting a single descriptor (LTP or LPQ) from the whole matrix/image; $\mathrm{ImL}_{\mathrm{LTP}}$ and $\mathrm{ImL}_{\mathrm{LPQ}}$ are obtained by dividing the image in ten blocks each of dimension $(N/10) \times 20$ and extracting the texture descriptors from each block; the resulting 10 vectors are finally concatenated for describing a given protein.

Local ternary patterns (LTP)

The LTP descriptor (Tan and Triggs 2007) is a variant of LBP (Ojala et al. 2002). Both the LBP and the LTP operators quantize the difference of each pixel x and its P neighboring pixels u_p on a circle of radius R around x. A histogram is then computed from these differences. LBP uses a binary encoding scheme that has the disadvantage of being sensitive to noise in the near-uniform regions. LTP solves this problem using 3 values instead of 2 to encode the difference between a pixel x and its neighbor u. The LTP operator is defined as follows:

The descriptor is constructed by transforming the ternary pattern into two binary patterns according to its positive and negative components. Two histograms are computed, one for the positive and one for the negative pattern, using a threshold value of 7. The two histograms are then concatenated. In this work, we report the performance obtained using the rotation invariant uniform bins and the uniform bins which we name $\mathrm{ImG}_{\mathrm{LTPu}}$, $\mathrm{ImG}_{\mathrm{LTPr}}$ and $\mathrm{ImL}_{\mathrm{LTPu}}$, $\mathrm{ImL}_{\mathrm{LTPr}}$, where subscript r stands for the rotation invariant uniform bins and subscript u stands for uniform bins. Our final descriptor has been obtained by concatenating the extracted features.[12] The features were obtained in the experimental section with settings ($P = 8$, $R = 1$) and ($P = 16$, $R = 2$).

Local phase quantization

The LPQ operator, first proposed in (Ojansivu and Heikkila 2008), is based on the blur invariance property of the Fourier phase spectrum. LPQ uses the local phase information extracted from the two-dimensional short-term Fourier transform (STFT) computed over a rectangular neighborhood defined by each pixel position. Only four complex coefficients are considered. They correspond to four fixed two-dimensional frequencies, which are separated into real and the imaginary parts and quantized as integers between 0 and 255 using a binary coding scheme [see (Ojansivu and Heikkila 2008) for mathematical details]. A histogram of these integer values is then computed and used as the feature vector. The histogram is normalized by dividing each element with the sum of the values of the histogram before it is used to train a classifier. In this work, we report the performance obtained using concatenated features extracted[13] by LPQ with radius 3 and

[11] Matlab code: http://www.cse.oulu.fi/CMV/Downloads/LPQMatlab.

[12] Implementing LTP from the matlab LBP code available at http://www.cse.oulu.fi/CMV/Downloads/LBPMatlab.

[13] Using the matlab code of LPQ available at http://www.ee.oulu.fi/mvg/download/lpq/.

Table 1 Summarized description of the datasets (if available the number of training and independent samples is given in column #Samples)

Name	References	Short name	#Samples	#Classes	Protocol
Membrane sub-cellular	(Chou and Shen 2007c)	MEM	3,249 + 4,333	8	Independent training and testing sets
DNA-binding proteins	(Guo et al. 2008)	DNA	349	2	Tenfold cross validation
Enzyme	(Nanni et al. 2009)	ENZ	1,094	2	Tenfold cross validation
GO dataset	(Nanni et al. 2009)	GO	168	4	Tenfold cross validation
Human interaction	(Pan et al. 2010)	HI	8,161	2	Tenfold cross validation
Submitochondria locations	(Du and Li 2006)	SL	317	3	Tenfold cross validation
IsEnzyme	(Lu et al. 2007)	IE	7,329	2	Tenfold cross validation
Virulent independent set 1	(Garg and Gupta 2008)	VI1	2,055 + 83	2	Independent training and testing sets
Virulent independent set 2	(Garg and Gupta 2008)	VI2	2,055 + 284	2	Independent training and testing sets

5. The Gaussian derivative quadrature filter pair is used for local frequency estimation.

Experiments

This section reports the results of an experimental evaluation of the protein descriptors with the aim of sequence-based protein classification performed on several datasets.

Datasets, testing protocols and performance indicators

The proposed approach has been evaluated on the following datasets, according to the testing protocols suggested by their creators. A brief summary description of each dataset and its testing protocol is reported in Table 1.

Membrane sub-cellular (MEM) (Chou and Shen 2007a): this dataset contains membrane proteins that belong to eight 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 goal of this dataset is to classify a given query protein in a given localization. All proteins in the same subcellular location have less than 80 % sequence identity. The testing protocol is based on a given subdivision, each of which is divided into training (3,249 proteins) and testing (4,333 proteins) sets.

DNA-binding proteins (DNA) (Guo et al. 2008): this dataset contains 118 dna-binding proteins and 231 non-DNA-binding proteins and have less than 35 % sequence identity between each pair.

Enzyme (ENZ) (Nanni et al. 2009): this dataset was created using the PDB archive and includes proteins annotated as enzymes, specifically 381 hydrolases and 713 enzymes of different kinds.

GO dataset (GO) (Nanni et al. 2009): this dataset was extracted from the PDB archive by selecting proteins according to GO annotations. It distinguishes the biological processes "immune response" (33 proteins) and "DNA repair" (43 proteins) and the molecular functions "substrate specific transporter activity" (39 proteins) and "signal transducer activity" (53 proteins). The presence of highly similar proteins within the same class was avoided by removing sequences which had more than 30 % identity.

Human interaction (HI) (Pan et al. 2010): The positive protein–protein interaction (PPI) dataset (Pan et al. 2010) was downloaded from the human protein references database (HPRD, June 2007 version). This version of HPRD contains 38,788 protein–protein pairs of experimentally verified PPIs from 9,630 different human proteins. Self-interactions and duplicate interactions from the dataset were eliminated to obtain 36,630 unique positive protein–protein pairs. The benchmark negative dataset was obtained from the Swiss-Prot database (version 57.3 released on 26 May 2009) by selecting 36,480 protein couples with different cellular compartments, each of which do not interact with each other [see (Pan et al. 2010) for details]. The final dataset was constructed from the original benchmark dataset by excluding proteins having more 25 % sequence identity to any of the other proteins. Screening was accomplished using the PISCES program. No protein pair in the remaining dataset has sequence identity higher than 25 %. In this way, the number of proteins in the positive dataset was reduced from 9,630 to 2,502, and the number of proteins in the negative dates was reduced from 2,184 to 661, for a total of 3,899 positive samples of protein pairs and 4,262 negative samples of protein pairs.

Submitochondria locations (SL) (Du and Li 2006): this dataset contains 317 proteins classified into three submitochondria locations: 131 inner membrane proteins; 41 outer membrane proteins; and 145 matrix proteins. Not more than 40 % similarity was allowed (i.e., the identity between any 2 sequences in the processed dataset had to be less than 40 %). This value was required to obtain a balance between the homologous bias and the size of the training set.

IsEnzyme (IE) (Lu et al. 2007): this dataset contains 2,443 enzymes and 4,886 non-enzyme proteins selected such that each protein shared not more than 25 % of identities with any other.

Virulent datasets 1 and 2 (VI1, VI2) (Garg and Gupta 2008): this dataset contains bacterial virulent protein sequences which were retrieved from the SWISS-PROT and VFDB (an integrated and comprehensive database of virulence factors of bacterial pathogens). The two independent sets share the same training set which consists of 1,025 virulent and 1,030 non-virulent bacterial sequences. The Virulent Independent dataset 1 (VI1) consists of 83 protein sequences, selected such that there are no two sequences that are more than 40 % similar. The Virulent Independent dataset 2 (VI2) consists of 141 virulent and 143 non-virulent sequences from bacterial pathogen sequences of organisms which were not represented in the training set.

In this work, we use three performance indicators: the classification accuracy, the area under the ROC curve (AUC) (Fawcett 2004) and the statistical rank. The accuracy is the ratio between the number of sample cases correctly classified and the total number of sample cases. The ROC curve is a graphical plot of the sensitivity of a binary classifier versus false positives (1 − specificity), as its discrimination threshold is varied; its area 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 and Duin 2007). The area under the ROC is considered one of the most reliable performance indicators (Qin 2006) as it is based on both sensitivity and specificity. The statistical rank reports the relative position of a method against the other tested: the average rank is the most stable indicator to average performance on different datasets, it is calculated using the Friedman's test (alpha = 0.05) applying the Holm post hoc procedure (Ulas et al. 2012).

Experimental results

The first experiment is aimed at comparing all the descriptors detailed in "Pattern representation and feature extraction" and summarized in Table 2, thus considering only stand-alone classification approaches.

In Tables 3 and 4, we report the accuracy and the AUC of each stand-alone approach.[14] From the results reported in the previous Tables, we can draw the following conclusions:

[14] The Tables 3 and 4 with standard deviations, for the datasets where a cross validation testing protocol is used, are available at: https://www.dropbox.com/s/k9szugy2at896j1/tabelleConSTD.docx.

Table 2 Summary of the descriptors

Descriptors		
Protein representation	Descriptor	Size
AAS	2G	400
	RC	1,200
	AC	40
PSSM/SMR	Matrix-based	
	AB	400
	ASC	400
	AM	300
	SA	400
	SA1	400
	PP	320
	Texture-based	640
	ImG_{LTPu}	
	ImG_{LTPr}	56
	ImG_{LPQ}	256
	ImL_{LTPu}	6,400
	ImL_{LTPr}	560
	ImL_{LPQ}	2,560

1. PP and SA1 are, on average, the best approaches using PSSM as protein representation;
2. PP and ImGLPQ are, on average, the best approaches using SMR as protein representation;
3. In large datasets (i.e. MEM and HI), the superiority of PP is not confirmed (mainly in the HI dataset);
4. RC outperforms AC in large datasets, while AC outperforms RC in the smaller dataset. Probably this is due to the high dimensionality of the feature vector extracted by RC that needs a large training set for avoiding the curse of dimensionality problem;
5. On average, PSSM representation outperforms SMR (which is based on the amino acid sequence), anyway SMR (also if only one physicochemical property is used) is comparable with the descriptors based on amino-acid sequence (RC and AC);
6. RC outperforms (or obtains similar performance) also PP extracted from the PSSM (PP^{PSSM}) representation of the proteins in large datasets but in the smaller ones PP^{PSSM} wins;
7. It is interesting to note that ImGLPQ (one of the novel descriptors here proposed) is the best approach among the tested descriptors extracted from PSSM and SMR representation in the HI dataset (the larger tested dataset), it also outperforms AC.

As a second experiment, we tested some ensemble approaches based on fusion of the following descriptors (Tables 5, 6) over all the nine datasets,:

8. *S2:* fusion by sum rule of RC and AC;

Table 3 Comparison among stand-alone approaches in terms of accuracy

Accuracy		Datasets									
Protein representation	Descriptor	MEM	DNA	ENZ	GO	HI	SL	IE	VI1	VI2	Rank
AAS	2G	89.2	82.0	45.0	39.1	92.1	67.4	67.3	73.1	68.5	11
	RC	90.7	82.9	45.7	39.4	95.6	70.7	70.3	75.9	70.8	9
	AC	88.4	86.8	44.6	50.6	90.2	82.9	69.7	73.1	69	4
PSSM	AB	92.1	78.5	51.7	51.9	91.4	81.0	68.2	73.2	67.5	18
	SA	88.8	87.9	43.2	55.6	90.2	83.9	80.2	72.3	78.5	3
	SA1	89.9	**92.1**	46.6	58.1	90.2	**87.1**	**81.6**	75.5	82.8	**1**
	ASC	91.2	84.1	47.6	46.9	93.0	83.2	79.2	74.1	79.5	7
	AM	84.0	80.0	44.5	44.4	94.1	77.4	73.2	64.7	75.1	14
	PP	**92.3**	88.5	**57.1**	**62.4**	88.6	**87.1**	81.5	75.9	**83**	2
	ImG_{LTPu}	84.2	78.8	45.8	49.4	81.5	77.1	73.2	66.6	73.2	16.5
	ImG_{LTPr}	89.0	78.2	45.6	43.1	94.4	78.7	74.0	65.2	74.1	19
	ImG_{LPQ}	89.5	84.7	46.5	51.3	93.2	81.6	74.1	67.5	76.8	6
	ImL_{LTPu}	84.9	79.7	46.0	48.8	81.4	77.1	72.2	65.2	75.1	15
	ImL_{LTPr}	79.7	75.3	42.8	33.8	69.9	42.6	60.5	58.5	52.3	21
	ImL_{LPQ}	82.1	75.9	44.0	33.8	81.4	42.6	59.8	59.0	56.8	20
SMR	AB	88.4	73.8	43.4	36.3	92.8	77.1	70.7	64.2	63.2	24
	SA	85.3	82.9	43.8	41.9	79.1	76.1	70.1	63.2	63.0	9
	SA1	85.3	82.9	43.8	41.9	79.1	76.1	70.0	62.8	62.9	9
	ASC	87.8	78.8	42.2	35.0	91.8	75.8	69.7	62.5	62.2	16.5
	AM	16.7	67.9	22.8	30.0	53.5	48.4	50.0	60.0	61.0	27
	PP	90.6	85.3	44.5	48.1	87.9	82.9	72.6	**77.1**	69.4	5
	ImG_{LTPu}	82.3	72.6	42.5	36.9	85.6	70.3	65.2	61.2	63.1	25
	ImG_{LTPr}	84.3	74.4	42.5	36.9	93.8	72.3	68.3	62.0	60.3	22
	ImG_{LPQ}	85.1	81.5	44.4	40.6	**94.2**	76.1	70.1	66.3	68.7	12.5
	ImL_{LTPu}	82.2	72.4	42.5	36.9	86.6	70.3	62.3	60.5	58.5	26
	ImL_{LTPr}	80.8	74.1	42.5	36.9	79.0	42.6	55.5	57.2	60.1	23
	ImL_{LPQ}	82.8	81.5	44.4	36.9	86.0	42.6	54.2	56.3	65.1	12.5

Bold values represent the best result in each dataset

9. *WS2:* Fusion by weighted sum rule among descriptors based on PSSM representation ($WS2^{PSSM} = 3 \times PP^{PSSM} + ImGLPQ^{PSSM}$), this ensemble is designed to show the usefulness of the *LPQ* descriptor coupled with PP both based on PSSM representation;
10. *WS3*: Fusion by weighted sum among RC, AC and PP extracted from the PSSM representation of the proteins ($WS3 = RC + AC + 3 \times PP^{PSSM}$);
11. *WS8*: Fusion by weighted sum among RC, AC (weight 1), PP extracted from the PSSM representation of the proteins (weight 4), PP extracted from the SMR representation of the proteins (weight 1), ImG_{LPQ} extracted from the PSSM representation of the proteins (weight 1), ImG_{LPQ} extracted from the SMR representation of the proteins (weight 1), SA1 extracted from the PSSM representation of the proteins (weight 2), SA1 extracted from the SMR representation of the proteins (weight 1); ($WS8 = RC + AC + 4 \times PP^{PSSM} + PP^{SMR} + ImG_{LPQ}^{PSSM} + ImG_{LPQ}^{SMR} + 2 \times SA1^{PSSM} + SA1^{SMR}$)

The weights and the selection of the methods of WS2, WS3 and WS8 were done by considering all the datasets used in the first experiment. When a large training set is available, it is also possible to tune the ensemble creation in order to optimize the performance. For example, if in HI we define an ensemble $WSHI = 2 \times RC + ImG_{LPQ}^{PSSM} + ImG_{LPQ}^{SMR}$ we obtain an accuracy of 95.91 and an AUC of 98.98.

In the following test, we show our motivation of the parameters setting of RC; in Table 7, we report the performance of RC with $m = 1$, RC $m = 2$, and RC $m = 3$ compared with the RC as used in the previous tests (sub-Sect. "Quasi residue couple"). It is clear that our concatenated descriptor RC is better the RC with a single value of m.

We can also compare our results with other recent state-of-the-art approaches where the same testing protocol is used. For example in (Chou and Shen 2007c), an accuracy of 91.7 % is reported in the MEM dataset, while our best is 94.14 % (WS3), while in HI dataset the best methods

Table 4 Comparison among stand-alone approaches in terms of AUC

AUC		Datasets									
Protein representation	Descriptor	MEM	DNA	ENZ	GO	HI	SL	IE	VI1	VI2	Rank
AAS	2G	94.0	87.0	68.3	64.8	93.4	81.1	70.0	84.2	74.8	11
	RC	**96.2**	86.7	65.5	65.7	98.9	82.9	71.1	**86.1**	76.3	12
	AC	93.6	91.2	66.7	72.2	95.9	92.6	73.5	85.1	76.8	4
PSSM	AB	93.4	77.3	73.5	73.9	96.4	89.9	82.1	77.5	82.2	22
	SA	93.9	93.0	68.0	75.9	95.7	**94.2**	87.0	79.2	86.2	3
	SA1	95.4	**95.3**	72.3	79.1	95.9	93.8	**87.3**	80.2	87.0	**1**
	ASC	95.8	90.7	66.9	68.6	97.5	93.0	85.6	78.9	84.8	5
	AM	90.9	84.7	68.0	68.7	**98.3**	86.0	78.5	75.2	78.5	15
	PP	96.0	95.0	**78.0**	**84.0**	94.8	94.2	87.2	80.6	**87.3**	2
	ImG_{LTPu}	91.7	85.9	62.4	72.9	89.4	86.6	78.2	75.3	79.1	13.5
	ImG_{LTPr}	93.6	89.5	66.8	65.4	98.3	90.6	79.0	76.8	84.2	10
	ImG_{LPQ}	93.4	90.5	66.8	74.1	97.6	90.7	79.2	76.1	83.9	8
	ImL_{LTPu}	92.1	85.9	62.9	72.9	88.8	86.6	77.8	76.0	82.1	13.5
	ImL_{LTPr}	89.5	80.5	58.0	44.8	92.3	44.3	64.2	69.0	68.2	21
	ImL_{LPQ}	93.4	83.0	62.0	51.3	93.5	50.5	65.2	70.0	71.8	19
SMR	AB	90.3	72.4	57.6	55.6	96.9	86.8	73.4	77.5	72.5	26
	SA	90.3	90.6	62.1	68.9	84.8	87.7	73.1	77.2	72.0	6.5
	SA1	90.3	90.6	62.2	68.9	84.8	87.7	73.1	77.4	69.9	6.5
	ASC	94.8	83.5	58.3	62.3	96.2	85.9	72.3	72.8	67.5	18
	AM	54.0	57.5	52.7	51.6	52.9	58.9	52.1	55.5	56.2	27
	PP	93.4	90.1	65.3	72.3	93.8	93.3	74.7	81.8	75.0	9
	ImG_{LTPu}	87.6	76.4	60.7	59.5	91.0	82.7	71.4	74.3	72.2	24
	ImG_{LTPr}	91.0	80.9	61.5	59.4	97.9	83.2	71.7	74.1	72.1	20
	ImG_{LPQ}	92.2	84.3	61.9	64.6	98.2	85.5	72.5	81.4	71.0	16
	ImL_{LTPu}	87.4	76.3	60.7	59.5	91.6	81.8	70.7	72.2	66.8	25
	ImL_{LTPr}	89.8	76.8	57.2	47.7	94.4	43.6	58.6	54.9	59.2	23
	ImL_{LPQ}	92.2	84.1	60.9	51.2	95.4	49.4	60.2	63.2	66.3	17

Bold values represent the best result in each dataset

Table 5 Comparison among ensembles and best stand-alone in terms of accuracy (with standard errors)

Accuracy	Datasets									
	MEM	DNA	ENZ	GO	HI	SL	IE	VI1	VI2	Rank
Best Stand-Alone										
RC^{AAS}	90.7	82.9 ± 1.44	45.7 ± 1.61	39.4 ± 3.85	**95.6** ± 0.33	70.7 ± 2.47	70.3 ± 1.62	75.9	70.8	10
AC^{AAS}	88.4	86.8 ± 2.12	44.6 ± 1.47	50.6 ± 4.61	90.2 ± 0.30	82.9 ± 1.93	69.7 ± 1.79	83.1	69	7
PP^{PSSM}	92.3	88.5 ± 1.67	57.1 ± 0.99	62.4 ± 3.10	88.6 ± 0.52	87.1 ± 2.45	81.5 ± 1.30	75.9	83	5
$SA1^{PSSM}$	89.9	**92.1** ± 1.71	46.6 ± 1.00	58.1 ± 3.36	90.2 ± 0.35	87.1 ± 2.68	**81.6** ± 1.85	75.5	82.8	6
PP^{SMR}	90.6	85.3 ± 1.70	44.5 ± 1.02	48.1 ± 4.57	87.9 ± 0.28	82.9 ± 1.52	72.6 ± 1.80	77.1	69.4	8
$ImGLPQ^{PSSM}$	89.5	84.7 ± 1.51	46.5 ± 1.05	51.3 ± 3.46	93.2 ± 0.27	81.6 ± 2.15	74.1 ± 1.72	67.5	76.8	9
$ImGLPQ^{SMR}$	85.1	81.5 ± 1.81	44.4 ± 0.78	40.6 ± 3.39	94.2 ± 0.20	76.1 ± 1.68	70.1 ± 1.43	66.3	68.7	11
Ensemble										
S2	90.8	85.3 ± 1.62	51.1 ± 1.38	49.4 ± 4.44	94.3 ± 0.28	76.8 ± 2.29	72.1 ± 1.59	84.3	70.8	3
WS2	93.3	90.9 ± 1.59	**59.8** ± 0.96	**62.5** ± 3.11	91.9 ± 0.26	87.4 ± 2.26	81.5 ± 1.44	77	83.1	2
WS3	93.1	91.5 ± 1.61	56.8 ± 0.97	61.9 ± 3.09	93.5 ± 0.25	**87.7** ± 2.23	79.2 ± 1.42	74.8	**83.8**	4
WS8	**94.1**	90.3 ± 1.57	56.2 ± 1.25	59.4 ± 3.23	93.1 ± 0.25	85.8 ± 2.13	81.5 ± 1.38	**85.5**	81.7	1

Where results were obtained from a single repetition (MEM, VI1, VI2), the standard error is not reported

Bold values represent the highest value in each dataset

Table 6 Comparison among ensembles best stand-alone in terms of AUC (with standard errors)

Best Stand-Alone	Datasets									
	MEM	DNA	ENZ	GO	HI	SL	IE	VI1	VI2	Rank
AUC										
RC^{AAS}	96.2	86.7 ± 1.61	65.5 ± 0.98	65.7 ± 2.85	**98.9** ± 0.24	82.9 ± 1.57	71.1 ± 1.74	86.1	76.3	10
AC^{AAS}	93.6	91.2 ± 1.34	66.7 ± 0.96	72.2 ± 2.61	95.9 ± 0.27	92.6 ± 0.71	73.5 ± 1.80	85.1	76.8	7
PP^{PSSM}	96	95 ± 1.11	78 ± 0.87	84 ± 1.82	94.8 ± 0.30	94.2 ± 0.55	87.2 ± 1.33	80.6	87.3	5.5
$SA1^{PSSM}$	95.4	95.3 ± 1.23	72.3 ± 0.98	79.1 ± 2.25	95.9 ± 0.28	93.8 ± 0.68	87.3 ± 1.22	80.2	87.0	5.5
PP^{SMR}	93.4	90.1 ± 1.45	65.3 ± 1.13	72.3 ± 2.79	93.8 ± 0.33	93.3 ± 0.63	74.7 ± 1.57	81.8	75.0	9
$ImGLPQ^{PSSM}$	93.4	90.5 ± 1.45	66.8 ± 1.26	74.1 ± 2.77	97.6 ± 0.27	90.7 ± 0.97	79.2 ± 1.45	75.1	83.9	8
$ImGLPQ^{SMR}$	92.2	84.3 ± 1.93	61.9 ± 1.06	64.6 ± 2.89	98.2 ± 0.25	85.5 ± 1.20	72.5 ± 1.73	81.4	71.0	11
Ensemble										
S2	96.2	93 ± 1.11	70.6 ± 0.96	72.4 ± 2.39	98.4 ± 0.26	90.8 ± 0.97	74.4 ± 1.61	**87.3**	77.2	4
WS2	96.5	95.7 ± 1.01	**79.4** ± 0.89	**84.4** ± 1.64	97 ± 0.24	94.3 ± 0.63	**87.6** ± 1.34	80.3	**87.9**	2.5
WS3	96.6	95.8 ± 0.98	78.9 ± 0.85	83.1 ± 1.66	98 ± 0.28	93.8 ± 0.53	87.2 ± 1.34	87.1	**87.9**	1
WS8	**96.8**	**95.9** ± 0.96	**79.4** ± 0.88	82.8 ± 1.63	97.6 ± 0.21	**94.9** ± 0.49	87.5 ± 1.33	87.1	87.5	2.5

Bold values represent the highest value in each dataset

Table 7 Comparison among different RC descriptors

AUC	Datasets									
Descriptor	MEM	DNA	ENZ	GO	HI	SL	IE	VI1	VI2	Rank
RC $m = 1$	95.3	85.0	60.6	64.0	96.5	78.1	69.3	84.5	74.3	4
RC $m = 2$	95.8	86.2	63.4	**66.1**	97.1	79.9	69.5	86.0	74.9	2
RC $m = 3$	**96.8**	85.8	62.0	65.3	97.0	79.5	70.8	85.7	75.6	3
RC	96.2	**86.7**	**65.5**	65.7	**98.9**	**82.9**	**71.1**	**86.1**	**76.3**	**1**

Bold values represent the highest value in each dataset

Table 8 Comparison with some state of the art approaches

Performance indicator	AUC			Accuracy	
Method	DNA	ENZ	HI	MEM	GO
Chou and Shen (2007c)	–	–	–	91.7	–
Pan et al. (2010)	–	–	98.20	–	–
Wang et al. (2012)	–	–	–	92.7	–
Nanni et al. (2009)	93.3	72.5	–	–	50.0
WS3	**95.9**	**79.4**	97.60	**94.1**	59.4
WS8	95.8	79.0	98.40	93.7	**62.5**
WSHI	–	–	**98.98**	–	–

Bold values represent the highest value in each dataset

reported in (Pan et al. 2010) obtains an accuracy of 96.4 and AUC of 98.20 which are comparable to *WS8*. In a recent paper (Wang et al. 2012), based on the fusion of PSSM-based features and pseudo amino acid composition, obtained in the MEM dataset an accuracy of 92.71 %. In Table 8, a comparison with some state-of-the-art approaches[15] is reported in terms of accuracy for the multi-class datasets and AUC for the two-class datasets.

In Table 9, we report some results obtained on the HI dataset unbalancing the training data (since the original dataset is almost perfectly balanced, which does not illustrate the true distribution of the data). The unbalancing of the training data has been obtained by undersampling the positive class to a ratio of "n/p" (n negative samples for p positive samples). The results reported in Table 9, which have been obtained as the average of ten experiments, are related to some stand-alone approaches and to the best ensemble for this dataset, i.e. : WSHI which is more robust than the stand-alone approaches with unbalanced training sets.

Finally, for a better comparison of the tested approaches on the different datasets, we report in Fig. 1a comparative

[15] Only results from methods that use the same testing protocol are reported (the results obtained by leave-one-out cross validation are not considered).

graph that shows in more detail which methods are good/bad overall or just for a specific dataset.

Finally, for a statistical validation of our experiments, we have used the Wilcoxon signed rank test (Demsar 2006) among each couple of methods in Table 10 reporting different p values ("–" indicates no statistical difference with a p value higher than 0.10). We have used the Bonferroni-Holm (Holm 1979) correction for multiple comparisons.

The most interesting results among those reported in Table 10 are:

1. WS8 outperforms with p values of 0.05 or 0.01 all the stand-alone methods;
2. WS2 outperforms with p values of 0.01 both the stand-alone methods that belong to that ensemble;
3. S2 outperforms with a p value of 0.05 both the stand-alone methods that belong to that ensemble.
4. Among the three proposed ensemble, there is not a clear winner.

Table 9 Comparison among methods in terms of AUC considering an unbalanced dataset

AUC	HI			
Method	Balanced	3/1	10/1	30/1
RC	98.90	98.50	91.50	89.65
ImG_{LPQ}^{SMR}	97.60	97.35	92.00	90.25
ImG_{LPQ}^{PSSM}	98.20	97.10	91.65	88.75
WSHI	**98.98**	**98.92**	**94.85**	**92.25**

Bold values represent the highest value in each dataset

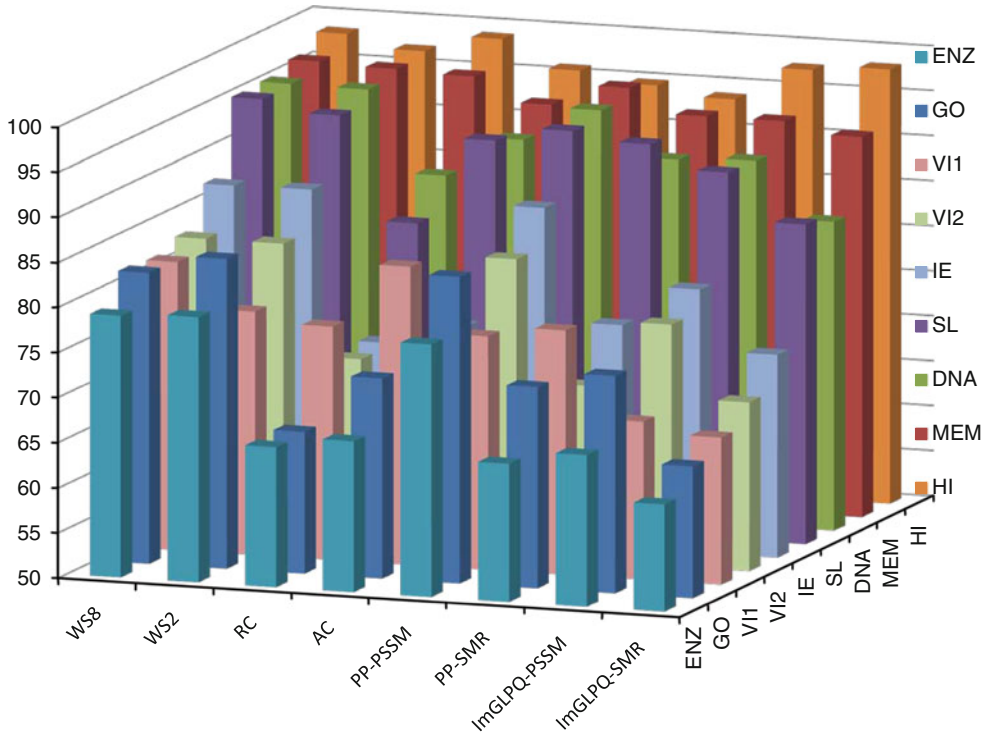

Fig. 1 Comparison among best approaches (AUC)

Table 10 Wilcoxon signed rank test among best approaches using AUC as performance indicator

	RC^{AAS}	AC^{AAS}	PP^{PSSM}	PP^{SMR}	ImG_{LPQ}^{PSSM}	ImG_{LPQ}^{SMR}	S2	WS2	WS3	WS8
RC^{AAS}	–									
AC^{AAS}	–	–								
PP^{PSSM}	0.05	0.05	–							
PP^{SMR}	–	–	0.05	–						
ImG_{LPQ}^{PSSM}	–	–	0.01	–	–					
ImG_{LPQ}^{SMR}	0.1	0.05	0.05	0.05	0.1	–				
S2	0.05	0.05	–	0.05	–	0.01	–			
WS2	0.05	0.05	0.01	0.01	0.01	0.01	0.1	–		
WS3	0.01	0.01	0.1	0.01	0.01	0.01	0.05	–	–	
WS8	0.05	0.01	0.05	0.01	0.01	0.01	0.05	–	–	–

Discussion

The results in Tables 2 and 3 show that there is not a representation which is superior to all the other in all the tested dataset. This fact, known as the "no free lunch" theorem in machine learning, means that no representation can be better than all others for all problems. To find the best for one problem you have to tune a system to the problem at hand. The results of our experiments can be viewed as an aid to make a choice on the base of the problem to be solved. In particular, we can draw the following conclusions:

1. PSSM is a good protein representation which works well mainly with PP and SA1 approaches;
2. Giving a texture representation to proteins and using a texture descriptor gives encouraging results: ImGLPQ is one of the best approaches using SMR as protein representation. The results of WS2 shown that it could be coupled with PP for improving its performance.
3. The size of datasets seems to play a main role in the choice of protein representation in fact the variation of performance among descriptors is stronger among in large datasets; in particular, approaches that use a high dimensional representation (e.g. RC) require larger dataset to avoid the curse of dimensionality. Moreover notice that when a large training set is available, it is possible to optimize the performance tuning the weights of a weighted sum rule, e.g. the weighted sum rule $2 \times \text{RC} + \text{ImG}_{\text{LPQ}}^{\text{PSSM}} + \text{ImG}_{\text{LPQ}}^{\text{SMR}}$ obtains an accuracy of 95.91 and an AUC of 98.98 in the HI dataset.
4. From the results reported Tables 4 and 5, we can conclude that designing an ensemble can be a smart method to improve the system performance: all the three reported ensembles outperform the stand-alone methods that built each ensemble. In particular, *WS2*, *WS3* and *WS8* work well, on average, on all the tested datasets and this is very useful for practitioners.

In the literature, there are several papers recently published based on the ensemble of descriptors, anyway most of them are tested on few datasets; in this work, differently from other recent papers, we propose an ensemble of descriptors/classifiers for sequence-based protein classification which works well on several datasets. Furthermore, instead of developing a web server, we share almost all the Matlab codes of the proposed approached (see the footnotes).

Conclusions

In this paper, we have presented an empirical study where different feature extraction approaches for representing proteins are compared and combined. Moreover, novel configurations based on PSSM and the substitution matrix are proposed for the first time and evaluated. Our experiments produced a number of statistically robust results regarding the generality and robustness of our system across an extensive evaluation of different datasets. Two main conclusions can be drawn from the results: (1) our proposed ensemble on average works well on all the tested datasets and would thus be very useful for practitioners; and (2) in the larger dataset (HI), ImG_{LPQ} (a novel approach proposed in this paper) is the best approach among the tested descriptors extracted from PSSM.

In this work we used a single SMR representation (i.e. SMR is created from a pre-fixed substitution matrix): one of the main advantages of SMR (not yet explored in this work) is the possibility to be defined starting from different substitution matrices. As a future work, we plan to design an ensemble to exploit the advantage of the fusion of different SMR representations.

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

For example, we have run some tests using a very different descriptor based on cloud of points (CP), see (Nanni 2005) for details, where from each protein patches of 30 amino acids are extracted. Each patch is described using AC and a one-class radial basis function SVM is trained for each protein. Finally the classifier profiles, see (Nanni 2005) for details, of each proteins are used to train LibSVM. CP obtains in the MEM dataset an accuracy of 0.84 % and an AUC of 0.91, when it is combined with WS3 it permits to improve its performance, i.e. the fusion $\text{CP} + \text{RC} + \text{AC} + 3 \times \text{PP}^{\text{PSSM}}$ obtains an accuracy of 0.94 and an AUC of 0.97 (in the MEM dataset).

Conflict of interest The authors declare that they have no conflict of interest.

References

Chou KC (2001) Prediction of protein cellular attributes using pseudo-amino acid composition. Proteins Struct Funct Genet 43:246–255

Chou KC (2009) Pseudo amino acid composition and its applications in bioinformatics, proteomics and system biology. Curr Proteomics 6:262–274

Chou KC, Shen HB (2007a) Review: recent progresses in protein subcellular location prediction. Anal Biochem 370:1–16
Chou KC, Shen HB (2007b) Signal-CF: a subsite-coupled and window-fusing approach for predicting signal peptides. Biochem Biophys Res Commun 357:633–640
Chou KC, Shen HB (2007c) MemType-2L: a Web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM. Biochem Biophys Res Commun 360:339–345
Chou KC, Shen HB (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(4):e9931
Chou KC, Zhang CT (1995) Review: prediction of protein structural classes. Crit Rev Biochem Mol Biol 30:275–349
Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge
Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
Du P, Li Y (2006) Prediction of protein submitochondria locations by hybridizing pseudo-amino acid composition with various physicochemical features of segmented sequence. BMC Bioinform 7:518
Fan GL, Li QZ (2011) Predicting protein submitochondria locations by combining different descriptors into the general form of Chou's pseudo amino acid composition. Amino acid (on-line press)
Fawcett T (2004) ROC graphs: notes and practical considerations for researchers. HP Laboratories, Palo Alto
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
Gribskov M et al (1987) Profile analysis: detection of distantly related proteins. Proc Nat Acad Sci USA 84:4355–4358
Guo J, Lin Y, Sun Z (2005) A novel method for protein subcellular localization: combining residue-couple model and SVM. In: Proceedings of 3rd Asia-Pacific Bioinformatics Conference, pp 117–129
Guo Y, Feng Y, Li M (2008) Predicting DNA-binding proteins: approached from Chou's pseudo amino acid composition and other specific sequence features. Amino Acids 34(1):103–109
Holm S (1979) A simple sequentially rejective multiple test procedure. Scand J Stat 6:65–70
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, California, pp 149–158
Jeong JC, Lin X, Chen X.-W (2011) On position-specific scoring matrix for protein function prediction. IEEE/ACM Trans Comput Biol Bioinform 8: 2
Kawashima S, Kanehisa M (1999) AAindex: amino acid index database. Nucleic Acids Res 27(1):368–369
Landgrebe TCW, Duin RobertPW (2007) Approximating the multiclass ROC by pairwise analysis. Pattern Recogn Lett 28(2007): 1747–1758
Lei Z, Dai Y (2005) An SVM-based system for predicting protein subnuclear localizations. BMC Bioinformatics 6:291
Leslie C, Eskin E, Noble WS (2002) The spectrum kernel: a string kernel for svm protein classification. Pacific Symposium on Biocomputing (PSB) 7:564–575
Leslie CS, Eskin E, Cohen A, Weston J, Noble WS (2004) Mismatch string kernels for discriminative protein classification. Bioinformatics 20:467–476
Li Yang, Yizhou Li, Rongquan Xiao, Yuhong Zeng, Jiamin Xiao, Fuyuan Tan, Menglong Li (2010) Using auto covariance method for functional discrimination of membrane proteins based on evolution information. Amino Acids 38:1497–1503
Lin WZ, Xiao X, Chou KC (2009) GPCR-GIA: a web-server for identifying G-protein coupled receptors and their families with grey incidence analysis. Protein Eng Des Sel 22(11):699–705
Lu L, Qian Z, Cai Y-D, Li Y (2007) ECS: an automatic enzyme classifier based on functional domain composition. Comput Biol Chem 31:226–232
Maddouri M, Elloumi M (2004) Encoding of primary structures of biological macromolecules within a data mining perspective. J Comput Sci Technol (JCST) 19(1):78–88
Nanni L (2005) Fusion of classifiers for predicting protein–protein interactions. Neurocomputing 68:289–296
Nanni L (2006) Comparison among feature extraction methods for HIV-1 protease cleavage site prediction. Pattern Recogn 39:711–713
Nanni L, Mazzara S, Pattini L, Lumini A (2009) Protein classification combining surface analysis and primary structure. Protein Eng Des Sel 22(4):267–272
Nanni L, Brahnam S, Lumini A (2010) High performance set of PseAAC and sequence based descriptors for protein classification. J Theor Biol 266(1):1–10
Ojala T, Pietikäinen M, Mäenpää T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24(7):971–987
Ojansivu V, Heikkila J (2008) Blur insensitive texture classification using local phase quantization. In: Lecture Notes in Computer Science 5099: 236–243 (ICISP)
Pudil P, Novovicova J, Kittler J (1994) Floating search methods in feature selection. Pattern Recogn Lett 5:1119–1125
Qin ZC (2006) ROC analysis for predictions made by probabilistic classifiers. Fourth International Conference on Machine Learning and Cybernetics 5:3119–3312
Rodriguez JJ, Kuncheva LI, Alonso CJ (2006) Rotation forest: a new classifier ensemble method. IEEE Trans Pattern Anal Mach Intell 28:1619–1630
Saidi R, Maddouri M, Nguifo EM (2010) Protein sequences classification by means of feature extraction with substitution matrices. BMC Bioinformatics 11:175
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 Eng Des Sel 20:39–46
Shen HB, Chou KC (2010) Gneg-mPLoc: a top-down strategy to enhance the quality of predicting subcellular localization of Gram-negative bacterial proteins. J Theor Biol 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
Wang JTL, Marr TG, Shasha D, Shapiro BA, Chirn GW (1994) Discovering active motifs in sets of related protein sequences and using them for classification. Nucleic Acids Res 22(14): 2769–2775
Wang J, Li Y, Wang Q, Zhang J, You X, Man J, Wang C, Gao X (2012) ProClusEnsem: predicting membrane protein types by fusing different models of pseudo amino acid composition. Comput Biol Med 42(5):564–574
Xiao X, Lin WZ (2009) Application of protein grey incidence degree measure to predict protein quaternary structural types. Amino Acids 37:741–749
Xiao X, Shao S, Ding Y, Huang Z, Huang Y, Chou KC (2005) Using complexity measure factor to predict protein subcellular location. Amino Acids 28:57–61

Xiao X, Shao SH, Huang ZD, Chou KC (2006a) Using pseudo amino acid composition to predict protein structural classes: approached with complexity measure factor. J Comput Chem 27(4):478–482

Xiao X, Shao SH, Ding YS, Huang ZD, Chou KC (2006b) Using cellular automata images and pseudo amino acid composition to predict protein subcellular location. Amino Acids 30:49–54

Xiao X, Wang P, Chou KC (2008a) Predicting protein structural classes with pseudo amino acid composition: an approach using geometric moments of cellular automaton image. J Theor Biol 254:691–696

Xiao X, Lin WZ, Chou KC (2008b) Using grey dynamic modeling and pseudo amino acid composition to predict protein structural classes. J Comput Chem 29:2018–2024

Xiao X, Wang P, Chou KC (2009a) GPCR-CA: a cellular automaton image approach for predicting G-protein-coupled receptor functional classes. J Comput Chem 30:1414–1423

Xiao X, Wang P, Chou KC (2009b) Predicting protein quaternary structural attribute by hybridizing functional domain composition and pseudo amino acid composition. J Appl Crystallogr 42:169–173

Xiao-Yong Pan, Ya-Nan Zhang, Hong-Bin Shen (2010) Large-scale prediction of human protein–protein interactions from amino acid sequence based on latent topic features. J Proteome Res 9:4992–5001

Yang ZR, Thomson R (2005) Bio-basis function neural network for prediction of protease cleavage sites in proteins. IEEE Trans Neural Netw 16:263–274

Yu X, Zheng X, Liu T, Dou Y, Wang J (2011) Predicting subcellular location of apoptosis proteins with pseudo amino acid composition: approach from amino acid substitution matrix and auto covariance transformation. Amino Acids 42(5):1619–1625

Zeng YH, Guo YZ, Xiao RQ, Yang L, Yu LZ, Li ML (2009) Using the augmented Chou's pseudo amino acid composition for predicting protein submitochondria locations based on auto covariance approach. J Theor Biol 259:366–372