Amino Acids (2011) 40:443–451
DOI 10.1007/s00726-010-0654-8

ORIGINAL ARTICLE

# Artificial intelligence systems based on texture descriptors for vaccine development

**Loris Nanni · Sheryl Brahnam · Alessandra Lumini**

Received: 27 December 2009 / Accepted: 3 June 2010 / Published online: 16 June 2010


**Abstract** The aim of this work is to analyze and compare several feature extraction methods for peptide classification that are based on the calculation of texture descriptors starting from a matrix representation of the peptide. This texture-based representation of the peptide is then used to train a support vector machine classifier. In our experiments, the best results are obtained using local binary patterns variants and the discrete cosine transform with selected coefficients. These results are better than those previously reported that employed texture descriptors for peptide representation. In addition, we perform experiments that combine standard approaches based on amino acid sequence. The experimental section reports several tests performed on a vaccine dataset for the prediction of peptides that bind human leukocyte antigens and on a human immunodeficiency virus (HIV-1). Experimental results confirm the usefulness of our novel descriptors. The matlab implementation of our approaches is available at http://bias.csr.unibo.it/nanni/TexturePeptide.zip.

**Keywords** Peptide classification · Vaccine development · HIV-1 protease prediction · Locally binary patterns · Discrete cosine transform · Support vector machine

L. Nanni (✉) · A. Lumini
Department of Electronic, Informatics and Systems (DEIS), Università di Bologna, Via Venezia 52, 47023 Cesena, Italy
e-mail: loris.nanni@unibo.it

A. Lumini
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

In most bioinformatics research, there are many classification problems related to peptides/proteins (e.g. subcellular localization (Chou and Shen 2007) and protein–protein interactions (Nanni and Lumini 2006a), it is important to search for and evaluate methods for reliably extracting features from peptides/proteins (Brusic et al. 2002).

In literature, most amino acid sequence descriptors are based on a vectorial representation. In nanni and Lumini (2006b), for example, the feature vector that describes a peptide is obtained by concatenating the vectors that describe each amino acid. This results in a 20-dimensional vector that represents each amino acid. The vector is all zeros except for the position corresponding to the considered amino acid, which takes on the value of a given physicochemical property. Another well-known method for extracting features from peptides/proteins is Chou's pseudo amino acid (PseAA) composition (Chou and Cai 2006) and its many variants, e.g. physicochemical distance (Chou 2000), digital code (Gao et al. 2005) and digital signal (Xiao and Chou 2007).

A completely different class of descriptors has been proposed, based on kernels. Some kernels used in amino acid representation include the Fisher kernel (Jaakkola et al. 1999), proposed for remote homology detection, the mismatch string kernel (Leslie et al. 2004), which has a performance similar to the Fisher kernel but is lower in computational cost, and a whole new class of kernels specifically proposed for predicting protein subcellular localization (Lei and Dai 2005).

To design useful vaccines for a large population, it is very important to predict the peptides that bind multiple human leukocyte antigen (HLA) molecules (Brusic et al. 2002). Developing automatic systems for predicting

whether a peptide binds multiple HLA molecules is very useful for making the design of vaccines more time effective. Examples of automatic systems that have been proposed in literature include systems based on support vector machines (SVMs) (Bozic et al. 2005), artificial neural networks (ANNs) and hidden Markov models (HMMs) (Brusic et al. 2004). For a good survey of automatic systems that predict whether a peptide binds multiple HLA molecules, see (Brusic et al. 2004).

The acquired immune deficiency syndrome (AIDS) virus is considered as one of the most devastating diseases humankind has ever faced, with more than 60 million people currently infected. It is well known throughout the scientific community that the HIV-1 protease is essential for the replication of the AIDS virus. A protease is an enzyme that cleaves proteins to their component peptides. The HIV-1 protease hydrolyzes viral polyproteins into functional protein products that are essential for viral assembly and subsequent activity. Inhibition of this protease prevents maturation of HIV particles and is thus a feasible way of blocking the viral life cycle. However, there is a major problem in synthesizing chemically modified inhibitors of the protease that can be used to bind the active site in HIV-1 protease: the protease cleaves at different sites with little or no sequence similarities.

HIV protease-susceptible sites in a given protein extend to an octapeptide region, the amino acid residues of which are sequentially symbolized by eight subsites, $P_4P_3P_2P_1P_1'P_2'P_3'P_4'$, and the counterparts in the HIV protease are symbolized by $S_4S_3S_2S_1S_1'S_2'S_3'S_4'$. According to the "lock and key" paradigm, if the amino acids in $P$ (the "key") fit the positions in $S$ (the "lock"), then the protease will cleave the octamer between positions $P_1$ and $P_1'$. Recently, several algorithms based on machine learning have been employed to learn this "lock and key" rule from a set of experimental observations. Studies specifically proposed to approach the HIV-1 protease problem using ANNs (specifically, feed-forward multilayer perceptrons) include Thompson et al. (1995), Cai and Chou (1998) and Narayanan et al. (2002). Better results have been obtained using linear SVMs (Rögnvaldsson and You 2003). Recently, a Web server (available at http://www.csbio.sjtu.edu.cn/bioinf/HIV/) has been created that predicts HIV-1 protease cleavage sites given a protein sequence (Shen and Chou 2008).

A bottleneck in the early development of machine learning techniques for this problem has been the lack of a large and reliable dataset that could be used to evaluate and compare existing approaches. One of the most commonly used datasets in the last 10 years, the HIV-1 PR 362 dataset (Rögnvaldsson and You 2003), has recently been shown to be unreliable (Rögnvaldsson et al. 2007). Rules on the most important peptides for the protease classification were discovered to be true only in HIV-1 PR 362. Some findings on the most selective physicochemical properties for classification extracted by Nanni and Lumini (2006b) from HIV-1 PR 362 were discovered not to be true for a newly developed dataset, the HIV-1 PR 1625 (Kontijevskis et al. 2007). Very recently, a new and larger dataset (the HIV-1 PR 3261) (Schilling and Overall 2008) has been collected. The most important published methods proposed in the bioinformatics literature are summarized in Table 1, where several brief indications are reported.

In this paper, we propose new techniques for using texture descriptors for representing a peptide. In particular, we compare two different methods for constructing a representation of the peptide as a matrix, and we use several variants of the discrete cosine transform (DCT) and the local binary pattern (LBP) descriptors. We find that the best results are obtained using 25 DCT coefficients with higher variance in the training data and by using a random subspace (RS) of 50 dominant local ternary patterns (LTP).

The remainder of this paper is organized as follows. In Sect. 2, we describe our system and introduce the peptide descriptors examined in this work. In Sect. 3, we report the experimental results. Finally, in Sect. 4 we draw a number of conclusions.

## System description

The aim of this work is to study the behavior of different peptide descriptors and their combination for HIV-1 protease cleavage site prediction and the predictions of peptides that bind HLA. All the approaches evaluated in the experimental section can be roughly schematized as in Fig. 1. Given a peptide represented as a fixed sequence of amino acids, feature extraction is performed to obtain a fixed length descriptor. Feature selection is then sometimes performed to optimize the feature extraction parameters. Finally, an SVM is used for classification. We describe steps 2 and 3 in more detail below.

## Feature extraction and selection

A peptide is a sequence of *Len* consecutive amino acid letters that can be numbered from 1, 2,…,20 according to the order of 20 native amino acids: A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V. Several fixed length descriptors for peptide representation include orthonormal representation (OR) (Rögnvaldsson and You 2003) and physicochemical encoding (PE) (Nanni and Lumini 2006b). We describe each of these below as well as the two texture descriptor (TD) matrix representations (Nanni and

**Table 1** The most important methods are summarized

| Application | Method | Novelty |
|---|---|---|
| Prediction of peptides that bind human leukocyte antigens | (Brusic et al. 2002) | It combines a novel representation of peptide/MHC interactions with a hidden Markov model as classifier |
| | (Brusic et al. 2004) | Some classifiers are compared, and the good performance of the neural networks is reported |
| | (Bozic et al. 2005) | It is shown that the best performance, starting from the descriptor proposed in Brusic et al. (2002), is obtained by SVM |
| | (Nanni and Lumini 2010) | It is shown for the first time that a peptide can be represented as a texture, and the texture features can be used for describing a given peptide |
| HIV protease classification | (Narayanan et al. 2002) | It reports that neural network performs better than decision tree as classifier |
| | (Rögnvaldsson and You 2003) | It reports than the HIV protease classification is a linear problem and that linear SVM performs better than neural networks |
| | (Nanni and Lumini 2006b) | An ensemble method based on the most selective physicochemical properties and linear SVM is proposed |
| | (Kontijevskis et al. 2007) | A larger dataset is collected, the HIV-1 PR 1625, and the problems of the previous most used dataset are detailed |
| | (Schilling and Overall 2008) | Another larger dataset, the HIV-1 PR 3261, is collected |
| | (Rögnvaldsson et al. 2009) | The generalization problem, i.e., the HIV-1 PR 1625 is used as training set and the HIV-1 PR 3261 is used as test set, is faced showing the good performance of the linear SVM |
| | (Nanni and Lumini 2009) | Several ensembles of classifiers are tested for the first time in this problem. It is shown that the ensembles outperform the stand-alone methods |

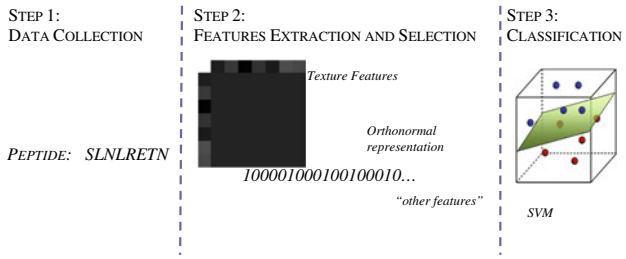


**Fig. 1** A general schema of the proposed approach

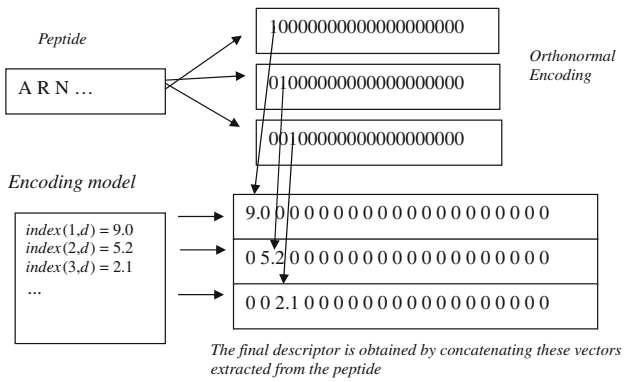


**Fig. 2** Example of physicochemical encoding representation

Lumini 2010): LBP (and its variants dominant LBP and LTP) and DCT.

In OR, each amino acid is mapped into a sparse orthonormal vector space using a 20-bit vector with 19 bits set to zero and the bit related to the amino acid position set to one. The peptide is represented by the concatenation of $Len \times 20$ features (Rögnvaldsson and You 2003).

In PE, each amino acid of the sequence is coded using a 20-dimensional vector with 19 values set to zero and the value related to the amino acid position set to the value of the fixed physicochemical property of the given amino acid (see Fig. 2) (Nanni and Lumini 2006b). The set of physicochemical properties are obtained from the amino acid index (Kawashima and Kanehisa 2000) database.[1] An amino acid index is a set of 20 numerical values representing any of the different physicochemical properties of amino acids. This database currently contains 544 such indices and 94 substitution matrices.[2] The number of possible encoding representations of each amino acid is given by the number NP = 544 + 94, which is the number of different physicochemical properties obtained by the amino acid index database[3] (Kawashima and Kanehisa 2000). An index function *index*(*j*, *d*) can be defined as a function that returns the value of the property *d* for the amino acid *j* ($d \in [1..NP]$, $j \in [1..20]$).

[1] Available at http://www.genome.jp/dbget/aaindex.html.

[2] Only the diagonal values are considered.

[3] Available at www.genome.jp/dbget/aaindex.html (Accessed 15 July 2009).

In TD, a matrix representation of a peptide is used to represent the information related to both the positions of the amino acids in the sequence and their physicochemical properties (Nanni and Lumini 2010). A feature extraction method is then applied.

We have examined two methods for representing a peptide as a matrix. The first is based simply on the physicochemical properties, where each element ($i$, $j$) of the texture that represents a peptide (given a physicochemical property $d$) is the following:

$$\mathrm{OM}_d(i,j) = index(a\min(i), d)) + index(a\min(j), d)$$

where $\mathrm{OM}_d \in \Re^{Len \times Len}$ is a squared matrix of the dimension of the peptide, $a\min(i)$ returns the index of the amino acid in position $i$, and $index(j, d)$ returns the value of the property $d$ for the amino acid $j$.

The second method for representing a peptide as a matrix is based on the Hasse matrix, reported in Nanni and Lumini (2010). The Hasse matrix representation can be calculated for each given physicochemical property $d$ according to the following procedure. The 20 amino acids are sorted according to the value of $d$, and a "ranking value" is assigned to each (Feng and Wang 2008). The first amino acid has value 1/20, the second 2/20, and so onto 1, as long as there are no two amino acids with the same values. Otherwise, the sequence is divided by the number of different values of $d$ in the sequence. For example, if the 20 bases are sorted in the following way according to $d$: $\mathrm{N} < \mathrm{K} < \mathrm{R} < \mathrm{Y} < \mathrm{F} = \mathrm{Q} < \mathrm{S} < \mathrm{H} < \mathrm{M} < \mathrm{W} < \mathrm{G} = \mathrm{L} < \mathrm{V} < \mathrm{E} < \mathrm{I} < \mathrm{A} < \mathrm{D} < \mathrm{T} < \mathrm{P} < \mathrm{C}$, then the corresponding ranking values are $rank_d(\mathrm{N}) = 1/18$, $rank_d(\mathrm{K}) = 2/18, \ldots, rank_d(\mathrm{C}) = 1$. Given a physicochemical property $d$ and a peptide, its representation matrix $\mathrm{OM}_d \in \Re^{Len \times Len}$ is a squared matrix of the dimension of the peptide, the elements of which are obtained as follows:

$$\mathrm{OM}_d(i,j) = \frac{1}{2}[rank_d(a\min(i)) + rank_d(a\min(j))]i,j \in [1..Len]$$

where the function $a\ \min(i)$ returns the index of the amino acid position.

The texture descriptors are extracted from $\mathrm{OM}_d$. The first texture descriptor used in our experiments, LTP (Tan and Triggs 2007), is a variant of LBP (Ojala et al. 2002), which is a texture operator defined by considering the binary difference between the value of a cell, $\mathbf{x}$, in a matrix, and the values of its $P$ neighborhood, placed on a circle of radius $R$. LBP is illustrated in Fig. 3. The resulting pattern associated with $\mathbf{x}$ is *uniform* if the number of transactions between "0" and "1" of the sequence is less than or equal to two. The LBP operator can be made rotation invariant by performing $P$-1 bitwise shift operations and selecting the smallest value. The final descriptor is the histogram that

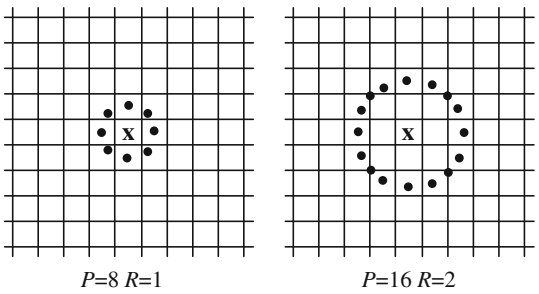


**Fig. 3** LBP neighborhood sets for different ($P$, $R$), where $P$ is the number of points and $R$ is the radius of the neighborhood

measures the occurrence of each type of uniform pattern and the number of non-uniform patterns contained in the whole matrix.

A problem with conventional LBP is that it is sensitive to noise in the near-uniform matrix regions. LTP overcomes this problem using a "trick" in the binarization process: in LTP the difference, $bd(\mathbf{x}, \mathbf{u})$, between a pixel $\mathbf{x}$ and its neighbor $\mathbf{u}$ is encoded by three values according to a threshold $\tau$:

$$bd(\mathbf{x},\mathbf{u}) = \begin{cases} 1 & \mathbf{u} \geq \mathbf{x} + \tau \\ 0 & \mathbf{x} - \tau \leq \mathbf{u} < \mathbf{x} + \tau 1 \\ -1 & \text{otherwise} \end{cases} \quad (1)$$

The ternary pattern is then split into two binary patterns by considering both its positive and negative components (see Fig. 4). Finally, the histograms that are computed from the binary patterns are concatenated to form the final descriptor. In this study, we have extracted and concatenated the histograms (as suggested in Ojala et al. (2002)) obtained using the following parameters: ($P = 8$; $R = 1$); ($P = 16$; $R = 2$), where $P$ is the number of points and $R$ is the radius of the neighborhood.

Liao et al. (2009) propose a method they call dominant LBP that utilize those LBP patterns that represent 80% of the whole pattern in the training data. In our experiments, we combine both dominant LBP and LTP descriptors.

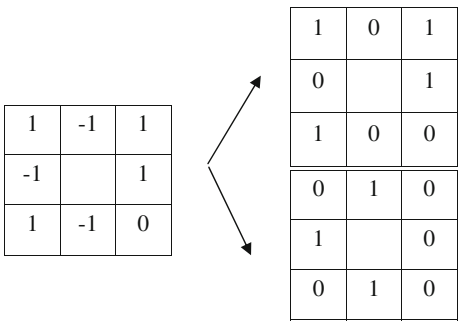


**Fig. 4** An example of splitting a ternary code into positive and negative LBP codes

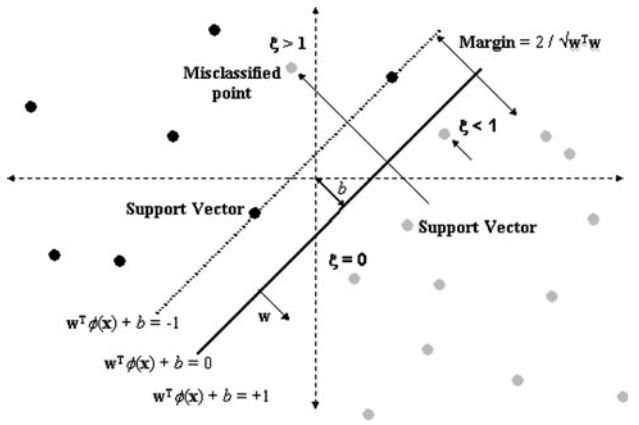


**Fig. 5** A graphical representation of the SVM hyperplane

The second texture descriptor used in our experiments is the DCT, as used in Nanni and Lumini (2010). DCT (Pan et al. 2000) is a texture descriptor that expresses a sequence of points in terms of a sum of cosine functions oscillating at different frequencies. DCT is similar to the discrete Fourier transform, except that it uses real numbers only. It has good information packing ability since most DCT components are typically very small in magnitude. This is because most of the salient information exists in the coefficients with low frequencies. As a result, when compared to other input independent transforms, DCT has the advantage of packing the most useful information into the fewest coefficients. In our experiments, we retain the DCT coefficients with highest variance considering only the training data.

### Classification

As illustrated in Fig. 1, the classifier used in our system is the SVM,[4] the same classifier used in Nanni and Lumini (2006b). The aim of SVM (Duda and Hart 1973) is to find the equation of a hyperplane that divides the training set into two classes, leaving all the points of one class on the same side while maximizing the distance between the two classes and the hyperplane. The basic two-class SVM formulation gets as input an implicit embedding $\Phi$ and a labeled training set $\{\mathbf{x}_i\}$ and returns the hyperplane $\mathbf{w}^T \Phi(\mathbf{x}) + b = 0$ that best separates the training samples of the two classes (see Fig. 5).

When the patterns that belong to the training set are not linearly separable, a different kernel [e.g. a polynomial kernel or radial basis function (RBF) kernel] can be used to map the input vectors into a higher dimensional feature space where an optimal hyperplane can be constructed.

[4] implemented as in OSU toolbox. www.ece.osu.edu/~maj/osu_svm/.

## Experimental results

In this section, we describe the results of our experiments using our proposed methods. We compare performance using our peptide descriptors on two databases: the vaccine and the HIV-1 PR 1,625 datasets.

In this section, the following two datasets are used for validating our proposed techniques:

- Vaccine (VAC) (Bozic et al. 2005), which contains peptides from five HLA-A2 molecules that bind/non-bind multiple HLA. The testing protocol suggested in Bozic et al. (2005) has been adopted, which is a "five-molecule" cross-validation method, where all the peptides related to a given molecule are used as the testing set and all the peptides related to the other four molecules form the training set (see Table 2 for details).
- HIV-1 PR 1625 dataset (HIV) (Kontijevskis et al. 2007), which contains 1,625 octamer protein sequences: 374 HIV-1 protease cleavable sites and 1,251 uncleavable sites. In this dataset, the tenfold cross-validation is used to assess the performance.

To compare the different approaches, we use the area under the receiver operating characteristic curve (AUC). This performance indicator is a scalar measure ranging between 0 (worst performance) and 1 (best performance). It can be interpreted as the probability that the classifier will assign a higher score to a randomly picked positive sample as opposed to a randomly picked negative sample (Qin 2006). The main advantage of using AUC instead of accuracy as the performance indicator is that it takes into account the scores of the classifiers.

In Tables 3 and 4, we compare the following methods:

- LBP/DCT/DAU, the methods based on LBP, DCT and Daubechies wavelet used in Nanni and Lumini (2010);
- SDC, the 25 DCT coefficients with higher variance in the training data;
- LBU, all the uniform patterns (not only the rotation invariant, as in LBP) are retained for training SVM;

**Table 2** Number of binders (B) and non-binders (NB) in training and testing sets for HLA-A2

| HLA-A2 | Training set | | Testing set | |
|---|---|---|---|---|
| | B | NB | B | NB |
| 0201 | 224 | 378 | 440 | 1,999 |
| 0202 | 619 | 2,361 | 45 | 25 |
| 0204 | 641 | 2,162 | 23 | 224 |
| 0205 | 648 | 2,346 | 16 | 40 |
| 0206 | 621 | 2,349 | 43 | 37 |

**Table 3** AUC obtained using the matrix representation of the peptide proposed in Nanni and Lumini (2010), based on the Hasse matrix. LBP, DCT and DAU as reported in Nanni and Lumini (2010)

| Dataset | | Descriptor from Hasse | | | | |
|---|---|---|---|---|---|---|
| | | LBP | DCT | SDC | DAU | LBU |
| HIV | *avg* | 0.803 | 0.890 | 0.895 | 0.723 | 0.814 |
| | *max* | 0.892 | 0.943 | 0.949 | 0.835 | 0.895 |
| VAC | *avg* | 0.788 | 0.800 | 0.817 | 0.667 | 0.820 |
| | *max* | 0.843 | 0.875 | 0.888 | 0.752 | 0.866 |

**Table 4** AUC obtained using the matrix representation proposed

| Dataset | | Descriptor from substitution matrix | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | LBP | DCT | SDC | DAU | LBU | DLB | DLT | RST |
| HIV | *avg* | 0.792 | 0.858 | 0.866 | 0.745 | 0.816 | 0.834 | 0.844 | 0.859 |
| | *max* | 0.861 | 0.935 | 0.933 | 0.813 | 0.890 | 0.881 | 0.916 | 0.927 |
| VAC | *avg* | 0.785 | 0.765 | 0.781 | 0.658 | 0.826 | 0.826 | 0.826 | 0.839 |
| | *max* | 0.827 | 0.830 | 0.849 | 0.701 | 0.864 | 0.852 | 0.868 | 0.875 |

- DLB, the dominant LBP;
- DLT, the dominant LTP with $\tau = 0.1$;
- RST, an RS of 50 DLT.

In the following tables, average (*avg*) and maximum (*max*) performance are obtained on the pool of physicochemical properties.

Since LBP and LBU obtain similar performance with the matrix representation of the peptide based on the Hasse matrix and with the representation based on the substitution matrix, and considering that the substitution matrix is easier to implement, we have extended our experiments to include some of the other LBP variants using the substitution matrix representation.

From the results reported in Tables 3 and 4, the following observations can be made:

- SDC, where the coefficients are selected using the variance, outperforms DCT.
- The variants of LBP that we tested outperform the standard rotation invariant uniform LBP (notice the performance difference between LBP and RST).
- It is clear that to optimize the neighborhood-based descriptors, the performance difference between LBP and RST, both based on the same idea, has to be seen. Notice that in this application problem, the neighborhood-based descriptors permit studying the correlation among neighbor amino acids (see the definition of LBP in Sect. 2).

In the methods described below, we named method X, H_X when the descriptor X is extracted from the peptide representation proposed in Nanni and Lumini (2010), based on the Hasse matrix, and we named it S_X when the descriptor X was extracted from the matrix representation we proposed.

**Table 5** Comparison among PE and some fusion approaches

| Dataset | | Descriptors | | | |
|---|---|---|---|---|---|
| | | PE | FUS1 | FUS2 | FUS3 |
| HIV | *avg* | 0.985 | 0.9252 | 0.983 | **0.986** |
| | *max* | 0.992 | 0.9646 | 0.990 | **0.993** |
| VAC | *avg* | 0.865 | 0.874 | 0.898 | **0.887** |
| | *max* | 0.884 | 0.908 | 0.910 | **0.897** |

The bold values represent the higest AUC in each dataset

In Table 5, we compare the following methods:

- PE, a standard method for extracting features from the amino acid sequence described in Sect. 2;
- FUS1, fusion by sum rule between H_SDC and S_RT;
- FUS2, fusion by sum rule among H_SDC, S_RST and PE;
- FUS3, fusion by weighted sum rule among H_SDC, S_RT and PE, the weight of H_SDC and S_RST is 1 while the weight of PE is 3.

In Table 6, we report the results obtained combining the different SVMs trained using 50 descriptors and by changing the physicochemical property used for building the matrix representation of the peptide. For example, the column SDC reports the AUC obtained combing 50 different SDC descriptors, each built using a different (randomly extracted) physicochemical property.[5] Those 50 SVMs (one for each SDC) are then combined using the sum rule. In Table 6, the results between parentheses is the average AUC obtained considering only one physicochemical property. As can be verified examining Table 6, a simple random selection of 50 physicochemical properties improves the performance of the texture descriptor representation.

The most interesting results reported in Tables 6 are:

- to combine descriptors based on different physicochemical properties is very useful in the texture-based approaches (see the AUC improvement of H_SDC and S_RST when, instead of a single physicochemical property, 50 descriptors are combined); instead, in the standard approach (i.e. PE), the performance improvement, when 50 descriptors are combined, is not so interesting. This is due to the consideration that different matrices are extracted from the same peptide using different physicochemical properties (see also Table 8 where the Q-statistic of H_SDC and S_RST, based on different physicochemical properties, is

[5] We have not considered the properties where the amino acids have a of value 0 or 1.

**Table 6** AUC obtained by combining with sum rule 50 descriptors each extracted from a different physicochemical property

| Dataset | Descriptors | | | | |
|---|---|---|---|---|---|
| | H_SDC | S_RST | PE | FUS1 | FUS3 |
| HIV | 0.973 (0.895) | 0.950 (0.859) | 0.985 (0.985) | 0.977 (0.925) | 0.988 (0.986) |
| VAC | 0.880 (0.817) | 0.861 (0.839) | 0.870 (0.865) | 0.917 (0.874) | 0.893 (0.887) |

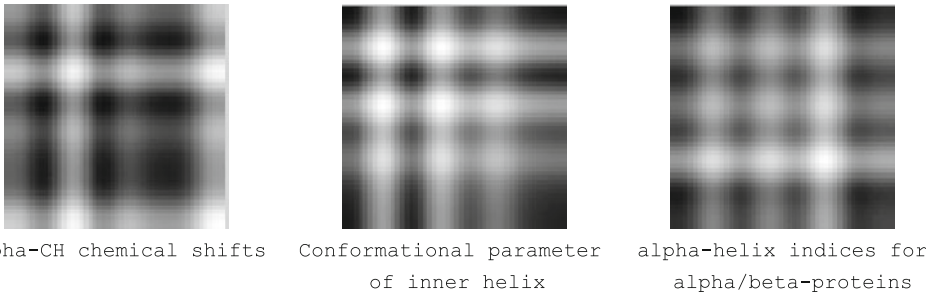


**Fig. 6** Some samples of textures extracted from the same peptide ('SLNLRETN') using different physicochemical properties (reported under the figures)

reported). In Fig. 6, we report some samples of texture extracted from the same peptide using different physicochemical properties for showing that different textures are extracted from the same peptide using different physicochemical properties.

- In the VAC dataset, the texture-based approach H_SDC outperforms PE; the method based on the weighted sum rule named FUS3 permits obtaining good performance in both HIV and VAC datasets. FUS3 obtains an accuracy of 97.0% in the HIV dataset and 82% in the VAC dataset.

Now we report, in Fig. 7, the specificity/sensitivity curve obtained by PE (the dark line) and our proposed method FUS3 (the gray line).

Also, these plots confirm our previous conclusions on the usefulness of combining our new approaches and a standard method based on the amino acid sequence.

As a further experiment, we investigated the relationship among the different feature extraction approaches tested in this paper by evaluating the error independence between the classifiers trained using those features. Table 7 reports the average Yule's Q-statistic (Kuncheva and Whitaker 2003) in the tested datasets for each couple of feature vectors. For two classifier $G_i$ and $G_j$, the Q-statistic a posteriori measure is defined as:

$$Q_{i,j} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}}$$

where $N^{ab}$ is the number of instances in the test set, classified correctly ($a = 1$) or incorrectly ($a = 0$) by the classifier $G_i$, and correctly ($b = 1$) or incorrectly ($b = 0$) by the classifier $G_j$. $Q$ varies between $-1$ and 1; $Q_{i,j} = 0$ for statistically independent classifiers. Classifiers that tend to recognize the same patterns correctly will have $Q > 0$, and those which commit errors on different patterns will have $Q < 0$. In this problem, the Q-statistic values are low enough (Kuncheva and Whitaker 2003) to validate the idea of combining the descriptors.

The results reported in Table 7 clearly show that both H-SDC and S-RST give different information with respect to PE. The main difference between PE and the texture-based approaches is that PE does not extract information on the correlation of neighbor amino acids (PE transforms the peptide in a vector representing each amino acid with its physicochemical value, see Sect. 2), while in the texture-based approaches each local neighborhood is given by a set of amino acids.

Moreover, in Table 8, we report the Q-statistic among different H-SDC and S-RST descriptors obtained using different physicochemical properties. It is clear that different physicochemical properties permit obtaining slightly different descriptors and for this reason their fusion permits improving the performance with respect to that obtained by a stand-alone descriptor (see Table 6).

As a further test, we report some results, see Table 9, to show how many observations are required to build a good predictor with the proposed encodings. We run the tests on the HIV dataset with a different number of training patterns (75, 50 and 25%). Ten experiments are performed (the training patterns are randomly extracted) and the average results are reported. From these results, it is clear that the new encoding needs the same number of training patterns of standard method as PE.

## Conclusion

In this paper, we have proposed new methods for describing the peptides, starting from the matrix representation of the peptides. Two different methods for

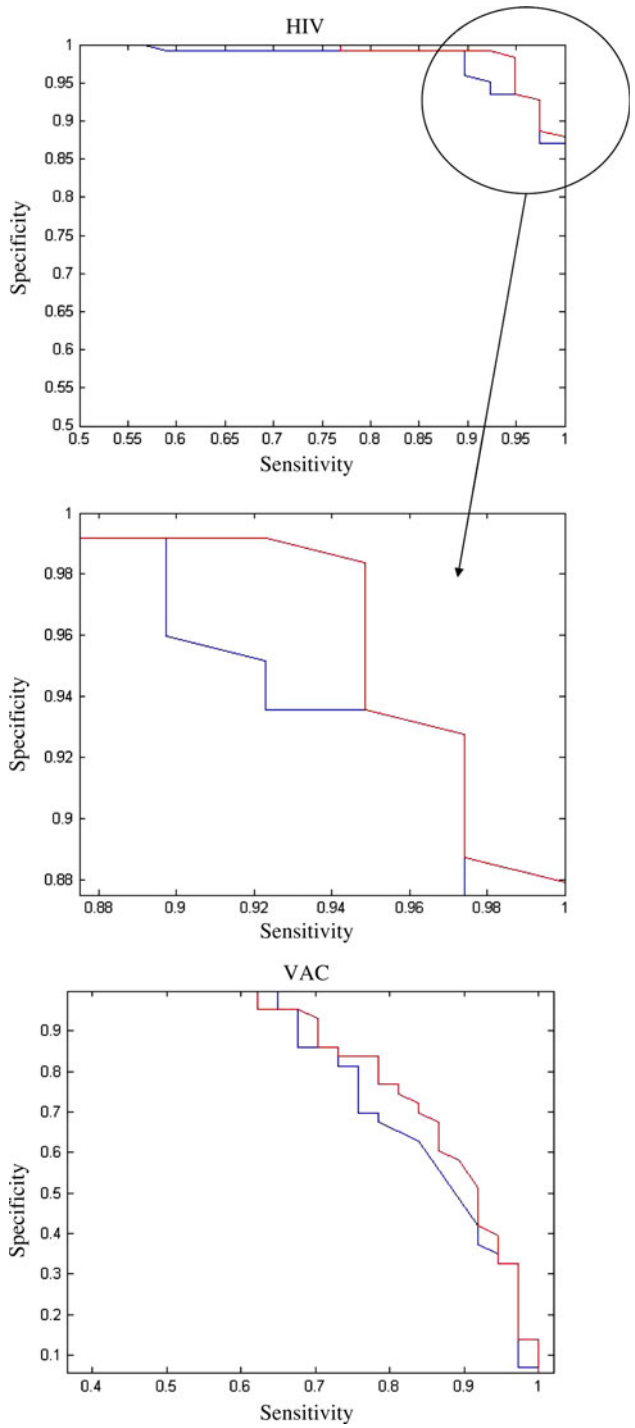


**Fig. 7** Specificity/sensitivity curve obtained by PE (the *dark line*) and our proposed method FUS3 (the *gray line*)

constructing a representation of the peptide as a matrix are compared, and different texture descriptors are extracted from these images.

Our tests on different datasets, HIV-1 protease cleavage site prediction and predictions of peptides that bind HLA, show that the proposed system outperforms previous methods based on a matrix representation of the peptides (Nanni and Lumini 2010). Moreover, fusion with the standard physicochemical encoding representation is studied. The best practical finding revealed in this work is that texture descriptors extracted from the matrix representation of the peptides and standard amino acid descriptors should be combined to obtain a more reliable method. Moreover, several tests using the Q-statistic are performed for studying the correlation among the tested methods. These results further confirm that texture-based and standard sequence amino acid-based methods provide different information.

In future, we would like to study a number of different texture descriptors, as well as better methods for combining those descriptors. Moreover, we would like to test our best approach FUS3 in other datasets for confirming the good performance of the weighted sum rule for combining texture-based and sequence-based peptide descriptors.

**Table 7** Yule's Q-statistic obtained using different peptide descriptors

| Feature extraction | | HIV | VAC |
|---|---|---|---|
| H-SDC | PE | 0.79 | 0.77 |
| S-RST | PE | 0.74 | 0.83 |
| H-SDC | S-RST | 0.65 | 0.67 |

**Table 8** Yule's Q-statistic among H-SDC and S-RST descriptors based on different physicochemical properties

| Feature extraction | HIV | VAC |
|---|---|---|
| H-SDC | 0.83 | 0.81 |
| S-RST | 0.84 | 0.81 |

**Table 9** AUC obtained in the HIV dataset with different numbers of training patterns

| Feature extraction | 75% | 50% | 25% |
|---|---|---|---|
| H-SDC | 0.971 | 0.962 | 0.926 |
| S-RST | 0.952 | 0.949 | 0.915 |
| PE | 0.985 | 0.979 | 0.967 |

## References

Bozic I, Zhang G, Brusic V (2005) Predictive vaccinology: optimization of predictions using support vector machine classifiers. Intell Data Eng Autom Learn LNCS 3578:375–381

Brusic V, Petrovsky N, Zhang G, Bajic VB (2002) Prediction of promiscuous peptides that bind HLA class I molecules. Immunol Cell Biol 80:280–285

Brusic V, Bajic VB, Petrovsky N (2004) Computational methods for prediction of T-cell epitopes a framework for modelling, testing, and applications. Methods 34:436–443

Cai YD, Chou KC (1998) Artificial neural network model for predicting HIV protease cleavage sites in protein. Adv Eng Softw 29:119–128

Chou KC (2000) Prediction of protein subcellular locations by incorporating quasi-sequence-order effect. Biochem Biophys Res Commun 278:477–483

Chou KC, Cai YD (2006) Predicting protein–protein interactions from sequences in a hybridization space. J Proteome Res 5:316–322

Chou KC, Shen HB (2007) 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

Duda RO, Hart PE (1973) Pattern classification and scene analysis. Academic Press, London

Feng J, Wang T-M (2008) Characterization of protein primary sequences based on partial ordering. J Theor Biol

Gao Y, Shao SH, Xiao X, Ding YS, Huang YS, Huang ZD, Chou KC (2005) Using pseudo amino acid composition to predict protein subcellular location: approached with Lyapunov index, Bessel function, and Chebyshev filter. Amino Acids 28:373–376

Jaakkola T, Diekhans M, Haussler D (1999) Using the Fisher kernel method to detect remote protein homologies. In: Seventh Int Conf Intell Syst Mol Biol, AAAI Press, Menlo Park, pp 149–158

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

Kontijevskis A, Wikberg JES, Komorowski J (2007) Computational proteomics analysis of HIV-1 protease interactome. Proteins Struct Funct Bioinform 1:305–312

Kuncheva LI, Whitaker CJ (2003) Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Machine learning, 51, pp 181–207

Lei Z, Dai Y (2005) An SVM-based system for predicting protein subnuclear localizations. BMC Bioinformatics, Dec 7, 6:291

Leslie CS, Eskin E, Cohen A, Weston J, Noble WS (2004) Mismatch string kernels for discriminative protein classification. Bioinformatics 20:467–476

Liao S, Law MWK, Chung ACS (2009) Dominant local binary patterns for texture classification. IEEE Trans Image Process 18:1107–1118

Nanni L, Lumini A (2006a) An ensemble of K-Local hyperplane for predicting protein–protein interactions. Bioinformatics 22:1207–1210

Nanni L, Lumini A (2006b) MppS: an ensemble of support vector machines based on multiple physicochemical properties of amino-acids. Neurocomputing 69:1688–1690

Nanni L, Lumini A (2009) Using ensemble of classifiers for predicting HIV protease cleavage sites in proteins. Amino Acids 36:409–416

Nanni L, Lumini A (2010) Coding of amino acids by texture descriptors. Artif Intell Med 48:43–50

Narayanan A, Wu X, Yang Z (2002) Mining viral protease data to extract cleavage knowledge. Bioinformatics 18:S5–S13

Ojala T, Pietikainen M, Maeenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Analysis Mach Intell 24:971–987

Pan Z, Rust A, Bolouri H (2000) Image redundancy reduction for neural network classification using discrete cosine transforms. In: Int Jt Conf Neural Netw, Como, Italy, pp 149–154

Qin ZC (2006) ROC analysis for predictions made by probabilistic classifiers. In: Fourth Int Conf Mach Learning Cybern, pp 3119–3312

Rögnvaldsson T, You L (2003) Why neural networks should not be used for HIV-1 protease cleavage site prediction. Bioinformatics, pp 1702–1709

Rögnvaldsson T, You L, Garwicz D (2007) Bioinformatic approaches for modeling the substrate specificity of HIV-1 protease: an overview. Expert Rev Mol Diagn 4:435–451

Rögnvaldsson T, Etchells TA, You L, Garwicz D, Jarman I, Lisboa PJ (2009) How to find simple and accurate rules for viral protease cleavage specificities. BMC Bioinformatics 16

Schilling O, Overall CM (2008) Proteome-derived, database-searchable peptide libraries for identifying protease cleavage sites. Nat Biotechnol 26:685–694

Shen HB, Chou KC (2008) HIVcleave: a web-server for predicting HIV protease cleavage sites in proteins. Anal Biochem 375:388–390

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

Thompson TB, Chou KC, Zheng C (1995) Neural network prediction of the HIV-1 protease cleavage sites. J Theor Biol 177:369–379

Xiao X, Chou KC (2007) Digital coding of amino acids based on hydrophobic index. Protein Pept Lett 14:871–875