Biological Sequence Classification with Multivariate String Kernels

Pavel P. Kuksa

Abstract—String kernel-based machine learning methods have yielded great success in practical tasks of structured/sequential data analysis. They often exhibit state-of-the-art performance on many practical tasks of sequence analysis such as biological sequence classification, remote homology detection, or protein superfamily and fold prediction. However, typical string kernel methods rely on analysis of discrete one-dimensional (1D) string data (e.g., DNA or amino acid sequences). In this work we address the multi-class biological sequence classification problems using *multivariate* representations in the form of sequences of features vectors (as in biological sequence profiles, or sequences of individual amino acid physico-chemical descriptors) and a class of *multivariate string kernels* that exploit these representations. On three protein sequence classification tasks proposed multivariate representations and kernels show significant 15-20% improvements compared to existing state-of-the-art sequence classification methods.

Index Terms—biological sequence classification, kernel methods

1 Introduction

Analysis of large-scale sequential data has become an important task in machine learning and data mining, inspired in part by numerous scientific and technological applications such as the biomedical literature analysis or the analysis of biological sequences. Classification of string data, sequences of discrete symbols, has attracted particular interest and has led to a number of new algorithms [1], [2], [3], [4], [5]. These algorithms often exhibit state-of-the-art performance on tasks such as protein superfamily and fold prediction [2], [4], [6], [7], or DNA sequence analysis [8].

A family of state-of-the-art kernel-based approaches to sequence modeling relies on *fixed length*, substring *spectral* representations of sequences and the notion of mismatch kernels, c.f. [2], [3]. There, a sequence is represented as the spectra (histogram of counts) of all short substrings (k-mers) contained within a sequence. The similarity score $K(X,Y)$ for pair of sequences X and Y is established by exact or approximate matches of k-mers contained in X and Y. Initial work, e.g., [3], [9], has demonstrated that this similarity can be computed using trie-based approaches in $O(k^{m+1}|\Sigma|^m(|X|+|Y|))$, for strings X and Y with symbols from alphabet Σ and up to m mismatches between k-mers. More recently, [10] introduced linear time algorithms with alphabet-independent complexity applicable to the computation of a large class of existing string kernels.

However, typical spectral models (e.g., mismatch/spectrum kernels, gapped and wildcard kernels [6], [3]) essentially rely on *symbolic Hamming-distance* based matching of 1D k-mers contained in the sequences. For example, given 1D sequences X and Y over alphabet Σ (e.g., amino acid sequences with $|\Sigma|$=20), the *spectrum-k* kernel [11] and the *mismatch-*(k,m) kernel [3] measure similarity between sequences as

$$K(X,Y|k,m) = \sum_{\gamma\in\Sigma^k} \Phi_{k,m}(\gamma|X)\Phi_{k,m}(\gamma|Y) = \sum_{\alpha\in X}\sum_{\beta\in Y}\sum_{\gamma\in\Sigma^k} I_m(\alpha,\gamma)I_m(\beta,\gamma) \quad (1)$$

where

$$\Phi_{k,m}(\gamma|X) = \left(\sum_{\alpha\in X} I_m(\alpha,\gamma)\right) \quad (2)$$

is the number of occurrences (possibly with up to m mismatches) of the k-mer γ in X, and $I_m(\alpha,\gamma) = 1$ if α is in the mutational neighborhood $N_{k,m}(\gamma)$ of γ, i.e. α and γ are at the Hamming distance of at most m. One interpretation of this kernel (Eq. 1) is that of cumulative Hamming distance-based pairwise comparison of all k-long substrings α and β contained in sequences X and Y, respectively. In the case of mismatch kernels the level of similarity of each pair of substrings (α,β) is based on the number of identical substrings their mutational neighborhoods $N_{k,m}(\alpha)$ and $N_{k,m}(\beta)$ give rise to, $\sum_{\gamma\in\Sigma^k} I_m(\alpha,\gamma)I_m(\beta,\gamma)$. For the spectrum kernel, this similarity is simply the exact matching of α and β.

We note that existing k-mer string kernels essentially use only *1D discrete* sequences (e.g., amino acid or nucleotide sequences) and Hamming-based matching.

However, as shown in previous works, using other, *multidimensional*, protein sequence representations is crucial in obtaining more accurate and robust predictions. In part, low sequence identities among distantly related proteins with similar structures and functions motivated the use of these other multidimensional amino-acid physico-chemical representations as physical and chemical properties of protein chains may preserve better

• *P. Kuksa is with NEC Laboratories America Inc, Princeton, NJ 08540. pkuksa@nec-labs.com*

among these otherwise very dissimilar primary protein sequences. For instance, 20-dim sequence profiles as in profile kernel method [2] or amino-acid descriptor vectors, e.g., as in recent work [12], [13], [14], can provide significantly more accurate results on a number of biological problems, including structural classification of proteins, protein remote homology detection [13], [14], protein function prediction [15], [16], protein subcellural localization [17], protein-protein binding prediction [18], [19], etc.

In cases of multidimensional sequence representations mentioned above (sequence profiles, or sequences of amino-acid descriptors), input data occurs in the form of sequences of *R-dim (real-valued) feature vectors* (i.e. *multivariate sequences*).

In this work, we consider an approach that directly exploits these richer R-dim multivariate sequences (sequence profiles and amino-acid descriptor sequences, in particular) and propose general, simple *discrete multivariate representations* of sequences (Sec. 3.1, 3.2). The proposed class of *multivariate similarity kernels* allows efficient inexact matching and classification of the multivariate discrete sequences (i.e. *sequences of R-dim discrete feature vectors*) (Sec. 3.3). The developed approach is applicable to both *discrete-* and *continuous*-valued original sequences, such as biological sequence profiles, or sequences of amino acid descriptors. Experiments using the new *multivariate* string kernels on protein remote homology detection and fold prediction show excellent predictive performance (Sec. 4) with significant *15%-20%* improvements in predictive accuracy over existing state-of-the-art sequence classification methods.

2 Related work

Over the past decade, various methods have been proposed to solve the sequence classification problem, including *generative*, such as HMMs, or *discriminative* approaches. Among the discriminative approaches, in many sequence analysis tasks, string kernel-based [20] methods provide some of the most accurate results [2], [3], [5], [6], [4], [21].

The key idea of basic string kernel methods is to map sequences of variable length into a fixed-dimensional vector space using feature map $\Phi(\cdot)$. In this space a standard classifier such as a support vector machine (SVM) [20] can then be applied. As SVMs require only inner products between examples in the feature space, rather than the feature vectors themselves, one can define a *string kernel* which computes the inner product in the feature space without explicitly computing the feature vectors:

$$K(X,Y) = \langle \Phi(X), \Phi(Y) \rangle, \tag{3}$$

where $X, Y \in D$, D is the set of all sequences composed of elements which take on a finite set of possible values from the alphabet Σ.

Sequence matching is frequently based on the co-occurrence of exact sub-patterns (k-mers, features), as in spectrum kernels [11] or substring kernels [22]. Inexact comparison in this framework is typically achieved using different families of mismatch [3] or profile [2] kernels. Both spectrum-k and mismatch-(k,m) kernels directly extract string features from the observed sequence, X. On the other hand, the profile kernel, proposed by Kuang et al. in [2], builds a $20 \times |X|$ profile [23] P_X and then uses a similar $|\Sigma|^k$-dimensional representation, now derived from P_X.

Most of existing string kernel methods essentially amount to analysis of 1D sequences over finite alphabets Σ with 1D k-mers as basic sequence features (as in e.g., spectrum/mismatch [11], [3], substring [22], gapped or wildcard kernels [6]). However, sequences can often be represented in the form of *sequences of feature vectors*, i.e. each input sequence X is a *sequence of R-dim feature vectors* which could be considered as $R \times |X|$ feature matrix (i.e. multivariate or *2D* sequence). For example, protein sequences could be considered as sequences of R-dim feature vectors (multivariate) describing physical/chemical properties of individual amino acids (e.g., as in [12]), or as *sequence profiles* (e.g., as in [2], [7]) describing each position as a probability distribution over amino acid residues.

A number of methods have been proposed previously that use amino-acid descriptors (feature vectors), including earlier works on protein function prediction (e.g., SVM-prot [16], ProtFun [24], DMP [15]), protein-peptide binding prediction [18], [19] using concatenation of 5-dim amino-acid descriptors as peptide representations, protein subcellular localization [17] using amino-acid kernel derived from amino-acid substitution matrices, prediction of protein folding class [25] using peptide-chain descriptors for composition and distribution of amino acid attributes, protein remote homology detection using AAindex attributes [14], [13] Recent work [12] uses BLOSUM descriptors to define k-mer similarity measure and shows improvements over traditionally used string kernels.

In contrast to previous work that either uses amino-acid feature vectors to obtain fixed-length global sequence descriptors (as in, e.g., [19], [25], [16]), or limits its use of amino-acid descriptors to define matching function for k-mers (e.g., [12], [17]), we propose novel methods that directly exploit these multivariate sequence representations (e.g., sequence of amino-acid descriptors, or sequence profiles) and allow to capture similarity/difference in *both* dimensions: along protein-chain, and along feature dimension (cumulative feature-chain similarity).

To this extent, we propose discrete and binary multivariate protein sequence representations (Sec. 3.1, 3.2) and consider a family of *multivariate* similarity kernels (Sec. 3, 3.3) that as we show empirically (Sec. 4) provide effective improvements in practice over traditional 1D (univariate) sequence kernels for a number of challeng-

ing biological problems, including protein remote homology detection, protein structural class classification, and protein binding prediction.

3 MULTIVARIATE DISCRETE SEQUENCE METHOD

In a typical setting, string kernels are applied to 1D (univariate) string data, such as amino acid sequences or DNA sequences. In this work we consider alternative *multivariate representations* for sequences (Fig. 1a) as *sequences of R-dim feature vectors* (e.g., sequences of amino-acid descriptor vectors, or amino-acid sequence profiles). In particular, given as input description of biological sequences in the form of sequences of (real-valued) identically sized amino-acid feature vectors, we consider the following two *discrete multivariate* representations:

1) *Symbolic embedding*. Encoding original real-valued R-dim feature vectors in discrete (binary) E-dim space using, e.g., similarity hashing approach [26] (Figure 1a; left subfigure);
2) *Direct feature quantization*. Directly quantizing each feature using, for example, uniform binning (Figure 1a; right subfigure), i.e. representing original (real-valued) $R \times |X|$ feature sequence as $R \times |X|$ discrete sequence.

In both approaches, the (real-valued) multivariate ($R \times |X|$) feature sequence X is re-represented as $E \times |X|$ or $|R| \times |X|$ multivariate (2D) discrete feature sequence. Figure 2 illustrates these representations for a given sequence X='SLFEQLGV'. In this figure, a 20-dim multivariate representation for original 1D sequence of amino acids is obtained by replacing each individual amino acid with a 20-dim vector of substitution BLOSUM62 scores which are then either directly discretized (Fig. 2a) or transformed into binary vectors using symbolic Hamming embedding (Fig. 2b). Resulting multivariate representations reflect underlying similarities among amino acids more accurately compared to symbolic 1D Hamming similarities.

We will show in the experiments that using these discrete *multivariate representations* can significantly (by *15-20%*) improve predictive accuracy compared to traditional 1D (univariate) kernel representations as well as other state-of-the-art approaches (Sec. 4).

In the following, we will discuss these proposed representation approaches in detail.

3.1 Direct feature quantization

In this approach, each feature $f^j, j = 1 \dots R$ is quantized by dividing its range (f^j_{min}, f^j_{max}) into a finite number of intervals. In the simplest case, the intervals can be defined, for instance, using uniform quantization, where the entire feature data range is divided into B equal intervals of length $\delta = (f_{max} - f_{min})/B$ and the index of quantized feature value $Q(f) = (f - f_{min})/\delta$ is used to represent the feature value f. Partitioning the feature data range could also be obtained by using 1D clustering, e.g. k-means, to adaptively choose dicretization levels.

In the experiments (Sec. 4) we use this approach with $20 \times |X|$ sequence profiles obtained from PSI-BLAST and $20 \times |X|$ BLOSUM substitution profiles, and show how using these multivariate representations with multivariate similarity kernels (Sec. 3.3) can improve the predictive ability of classifiers on a number of protein sequence classification tasks.

3.2 Discrete (symbolic) Embedding

Given multivariate input sequence $X = x_1, \dots, x_n$ of R-dim feature vectors, each R-dim vector can be mapped into discrete feature vectors using symbolic embedding $E(\cdot)$ as in, for example, similarity hashing [26]. Using similarity hashing, input sequence $X = x_1, ..., x_n$ of R-dimensional feature vectors, $x_i \in \mathcal{R}^R \quad \forall i$, is mapped intro a *binary* Hamming-space embedded sequence

$$E(X) = E(x_1), ..., E(x_n),$$

where $E(x_i) = e^i_1 e^i_2 \dots e^i_B$ is a symbolic Hamming embedding for item x_i in X, with $|E(x_i)| = B$, the number of bits in a resulting binary embedding of x_i. This embedding as proposed in [26] essentially aims to minimize average Hamming distance between binary embeddings corresponding to similar R-dim data points:

$$\min \sum_{\alpha,\beta} S(\alpha, \beta) h(E(\alpha), E(\beta))^2 \tag{4}$$

where $S(\alpha, \beta)$ is the similarity between data points α and β in the original R-dim space and $h(E(\alpha), E(\beta))$ is the *Hamming* distance between *binary* vectors $E(\alpha)$ and $E(\beta)$ in the Hamming embedded space.

The solution (set of embedding binary vectors $E(\cdot)$ that minimize the embedding objective in Eq. 4) can be obtained by solving the eigenvalue problem as shown in [26] and thresholding eigenfunctions to obtain binary codes.

Under this binary Hamming embedding, the Hamming similarity, $h(E(\alpha), E(\beta)$, between two B-dim feature embeddings $E(\alpha)$ and $E(\beta)$ is proportional to the original similarity score $\mathcal{S}(\alpha, \beta)$ between R-dim vectors α and β, i.e.

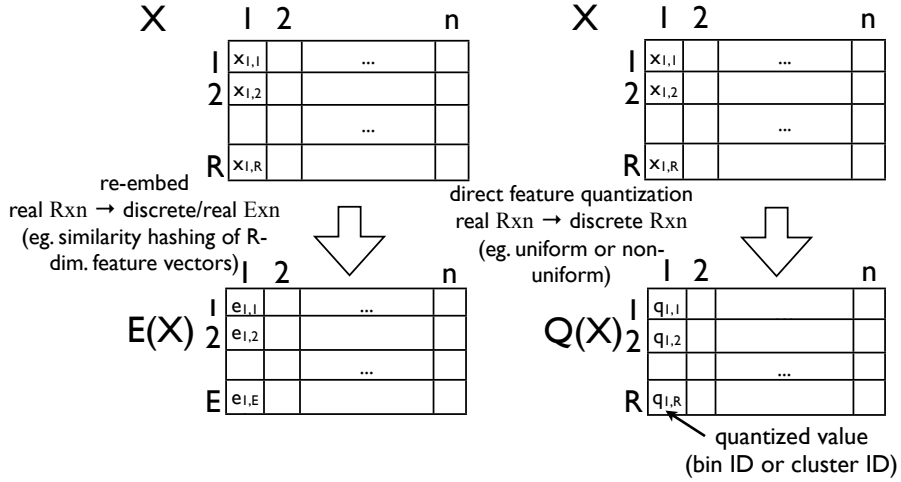
$$h(E(\alpha), E(\beta)) \propto \mathcal{S}(\alpha, \beta). \tag{5}$$

Using this Hamming embedding approach, original $R \times |X|$ (real-valued) feature sequence X is represented as $E \times |X|$ *binary* feature sequence, which can then be used with the string kernel method.

3.3 Multivariate similarity kernels

We now introduce efficient *multivariate similarity kernels* for the discrete multivariate sequence representations defined in Sec. 3.

The input sequences X and Y are sequences of identically sized (e.g., discrete R-dim or binary B-dim) amino-acid feature vectors (see Fig. 2). Similarity evaluation



(a) Proposed approach. Input sequence X of R-dim feature vectors is represented in 2D using direct feature quantization $Q(X)$ or embedding $E(X)$. Multivariate string kernel is used to measure sequence similarities.

Fig. 1: Proposed discrete *multivariate* representations.

Input seq. X: S L F E Q L G V

A	5	3	2	3	3	3	4	4
R	3	2	1	4	5	2	2	1
N	5	1	1	4	4	1	4	1
D	4	0	1	6	4	0	3	1
C	3	3	2	0	1	3	1	3
Q	4	2	1	6	9	2	2	2
E	4	1	1	9	6	1	2	2
G	4	0	1	2	2	0	10	1
H	3	1	3	4	4	1	2	1
I	2	6	4	1	1	6	0	7
L	2	8	4	1	2	8	0	5
K	4	2	1	5	5	2	2	2
M	3	6	4	2	4	6	1	5
F	2	4	10	1	1	4	1	3
P	3	1	0	3	3	1	2	2
S	8	2	2	4	4	2	4	2
T	5	3	2	3	3	3	2	4
W	1	2	5	1	2	2	2	1
Y	2	3	7	2	3	3	1	3
V	2	5	3	2	2	5	1	8

(a) Representation of protein sequence X using direct feature quantization. Each amino acid is represented using 20-dim vector of BLOSUM62 substitution probabilities (high values indicate substitutions that are more likely).

Input seq. X: S L F E Q L G V

1	1	0	0	1	1	0	1	0
2	1	1	0	1	1	1	1	0
3	0	0	0	0	0	0	1	0
4	0	1	0	1	1	1	0	1
5	0	1	1	1	0	1	1	1
6	1	1	1	1	1	1	1	1
7	1	1	1	0	0	1	1	1
8	1	1	1	1	1	1	1	1

(b) Representation of protein sequence X using symbolic binary embedding $E(X)$. Each amino acid is represented as an 8-bit binary vector obtained using similarity hashing. Note that similar amino acid have similar binary representations (e.g., hydrophobic amino acids L, and V or hydrophilic amino acids E, and Q).

Fig. 2: Examples of discrete *multivariate* representations for protein sequences using direct feature quantization or similarity hashing (binary Hamming embedding).

between the two sequences X and Y under *multivariate (2D) representations* amounts to comparing pairs of $k \times R$ ($k \times B$) *2D submatrices* contained in X and Y, where k is the length of 2D-k-mer (i.e. k is similar to the length of the k-mer in typical k-mer based univariate kernels). A *multivariate* string kernel can then be defined for the multivariate (R-dim or B-dim) sequences X and Y as

$$K_{2D}(X,Y) = \sum_{\alpha_{2D} \in X} \sum_{\beta_{2D} \in Y} \mathcal{K}(\alpha_{2D}, \beta_{2D}) \qquad (6)$$

where α_{2D} and β_{2D} are $|R| \times k$ (or $|E| \times k$) submatrices (2D=k-mers) of X and Y and $\mathcal{K}(\alpha_{2D}, \beta_{2D})$ is a kernel function defined for measuring similarity between two submatrices. The multivariate kernel in Eq. 6 corresponds to cumulative pairwise comparison of submatrices (2D-k-mers) contained in multivariate sequences X and Y (this is similar to typical spectrum kernels, e.g. mismatch kernel in Eq. 1).

One possible definition for the submatrix kernel $\mathcal{K}(\cdot,\cdot)$ in Eq. 6 is *row-based similarity* function

$$\mathcal{K}(\alpha_{2D}, \beta_{2D}) = \sum_{i=1}^{R} I(\alpha_{2D}^i, \beta_{2D}^i) \qquad (7)$$

where $I(\cdot,\cdot)$ is a similarity/indicator function for match-

ing 1D rows α^i_{2D} and β^i_{2D}. The matching function $I(\cdot,\cdot)$ could be defined as $I(\alpha,\beta)=1$ if the Hamming distance $d(\alpha,\beta) \leq m$, and 0 otherwise (i.e. similar to the mismatch kernel). In the experiments, we use spectrum, mismatch [3], and spatial sample (SSSK) [4] kernel matching functions as our 1D row matching function $I(\alpha^i_{2D},\beta^i_{2D})$ in Eq. 7, which results in corresponding *multivariate* spectrum, mismatch, and spatial sample kernels (referred as 2D-Spectrum, 2D-Mismatch, and 2D-SSSK, respectively).

Intuitively, according to the kernel definition (Eq. 7), similar submatrices (2D-k-mers) (i.e. submatrices with many similar rows) will result in high kernel value $\mathcal{K}(\cdot,\cdot)$.

Using Eq. 7, the multivariate (2D) kernel in Eq. 6 can be written as

$$K_{2D}(X,Y)=\sum_{i=1}^{R}\sum_{\alpha_{2D}\in X}\sum_{\beta_{2D}\in Y} I(\alpha^i_{2D},\beta^i_{2D}) \qquad (8)$$

which can be efficiently computed by running spectrum kernel with 1D k-mer matching function $I(\cdot,\cdot)$ R times, i.e. for each row $b=1\ldots R$. The overall complexity of evaluating multivariate kernel $K_{2D}(X,Y)$ for two multivariate (R-dim) sequences X and Y is then $O(R\cdot k\cdot n)$, i.e. is linear in sequence length n.

4 Experimental evaluation

We study the performance of our methods in terms of predictive accuracy on a number of challenging biological problems using standard benchmark datasets for protein sequence analysis.

4.1 Datasets and experimental setup

We test proposed methods on a number of multi-class biological sequence classification and prediction tasks:

(1) Protein remote homology detection task. This tasks tests the ability to build a classifier that would correctly recognize proteins from previously unseen protein families belonging to the target superfamily. We use two benchmark datasets (SCOP 1.53 and SCOP 1.59) that have been used by many previous works (e.g., [27], [2], [13], [28]). These benchmark datasets are derived from SCOP (Structural Classification of Proteins) database which aims to categorize proteins into structural hierarchy (Fig 3) of classes, folds, superfamilies, and families.

(2) Protein fold recognition task. The task here is to correctly recognize proteins from previously unseen superfamilies under target protein fold. We use a dataset with 4671 sequences derived from SCOP (SCOP 1.73) to simulate fold recognition problem.

(3) Multi-class protein fold classification. The task here is to classify given protein into correct fold. For this task, we use two benchmark datasets. The first dataset is a benchmark dataset (Ding-Dubchak dataset) containing 694 protein sequence from 27 protein folds [29], [7]. The second dataset is a larger dataset with 3860 protein sequences from 26 different folds [7]. For both datasets, the protein sequences are divided into training and testing sets [29], [7].

(4) MHC-peptide binding prediction. The goal of this prediction task is to predict whether a given peptide would bind to the target major histocompatibility complex (MHC) protein molecule. We use the benchmark dataset proposed in [30] for this task.

We provide details of the tasks and benchmark datasets in Table 1. For each of the tasks, we now provide details of datasets, experimental settings, train/test procedures.

4.1.1 Protein remote homology detection dataset

For the remote protein homology prediction task, we follow standard experimental setup used in previous studies [27] and evaluate average classification performance (ROC50) on 54 remote homology experiments, each simulating the remote homology detection problem by training a classifier on a subset of families (positive examples) under the target superfamily and testing the superfamily classifier on the remaining (held-out) families from the target superfamily according to SCOP hierarchy (Figure 3). Negative training and testing examples are chosen from protein families outside of target superfamily fold [27], [11]. For example, as shown in the figure, the two families (1.a.1.1 and 1.a.1.2) from the target superfamily 1.a.1 will be used as positive training data, while the held-out sequences from the third family (1.a.1.3) will be used for testing classifier's performance.

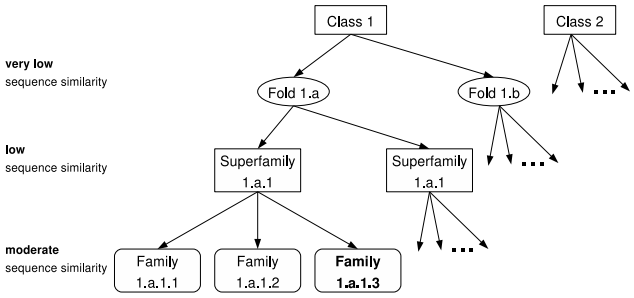


Fig. 3: Structural Classification of Proteins (SCOP) hierarchy. Protein domains organized into classes, folds, superfamilies, and families. Protein sequences from the same superfamily but different families are considered remote homologs.

4.1.2 Protein fold recognition dataset

Ding et al. [29] designed a challenging fold recognition data set [1] which has been used as a benchmark in many studies, for example [7]. The data set contains sequences from 27 folds divided into two *independent* sets, such that the training and test sequences share less than 35% sequence identities and within the training set, no sequences share more than 40% sequence identities.

1. http://ranger.uta.edu/~chqding/bioinfo.html

TABLE 1: Remote homology and fold prediction datasets

Dataset	No. seqs	No. classes	Training	Testing	Evaluation	Source
Protein remote homology (SCOP 1.59)	7329	54	-†	-†	54 binary train/test splits	[27]
Protein remote homology (SCOP 1.53)	4352	52	-†	-†	54 binary train/test splits	[27]
Protein fold detection (SCOP 1.73)	4761	96	-†	-†	96 binary problems	
Protein fold classification (D&D dataset)	694	27	311	383	multi-class train/test split	[29]
Protein remote fold classification	3860	26	3246	614	multi-class train/test split	[7]

† number of training/testing examples varies per experiment (i.e. depends on target fold/superfamily)

4.1.3 Remote protein fold detection dataset

Melvin et al. [7] derived this data set from SCOP 1.65 [31] for the tasks of multi-class remote fold detection. The data set contains 26 folds, 303 superfamilies and 652 families for training with 46 superfamilies held out for testing to model remote fold recognition setting.

4.1.4 MHC-peptide binding prediction dataset

The task here is to predict which peptides bind to a given major histocompatibility complex (MHC) protein molecule. Each type of MHC molecules (MHC alleles) have unique binding preferences and as a result the repertoire of the binding peptides varies among various MHC alleles. The task of predicting whether the peptide is a binder or a non-binder for a given MHC allele is important task in immunoinformatics and clinical research. We use the IEDB benchmark dataset [30] which contains quantitative binding data (IC_{50} values) for short peptides (9-mers) with respect to various MHC alleles. Peptides with $IC_{50} \geq 500$ are considered to be *non-binding*, while peptides with $IC_{50} < 500$ are considered to be *binding*.

4.2 Baseline methods

We compare our approach with a number of other state-of-the-art methods for sequence classification, both in fully supervised and semi-supervised settings, including spectrum/mismatch [11], [3], spatial sample kernels [4], substitution [6], and semi-supervised profile kernels [2].

The spectrum/mismatch kernel for two sequences X and Y corresponds to cumulative pairwise comparison of k-mers contained in X and Y

$$K(X,Y) = \sum_{\alpha \in X} \sum_{\beta \in Y} \mathcal{S}_m(\alpha, \beta),$$

where the k-mer matching/similarity function

$$\mathcal{S}_m(\alpha,\beta) = \sum_{\gamma \in \Sigma^k} I_m(\gamma,\alpha) I_m(\gamma,\beta)$$

indicates the number of k-mers γ at Hamming distance of at most m from both α and β ($I_m(\alpha,\beta) = 1$ if the Hamming distance $h(\alpha,\beta) \leq m$, and 0, otherwise).

The substitution kernel [6] replaces Hamming distance-based indicator function $I_m(\alpha,\beta)$ with a scoring function

$$I_\sigma(\alpha,\beta) = \begin{cases} 1, -\sum_{i=1}^{k} \log p(\alpha_i|\beta_i) < \sigma \\ 0, otherwise \end{cases}$$

($p(\alpha_i|\beta_i)$ is a conditional substitution probability).

The profile kernel [2]

$$Kp(X,Y) = \sum_{\gamma \in \Sigma^k} \sum_{i_X=1}^{|X|-k+1} \sum_{i_Y=1}^{|Y|-k+1} I_p(p_{i_X}(X),\gamma) I_p(p_{i_Y}(Y),\gamma)$$

uses *position-specific* scoring function $I_p(p_i,\beta)$ over $20 \times |X|$ profile $p(X) = p_1 \dots p_{|X|}$

$$I_p(p_i,\beta) = \begin{cases} 1, -\sum_{j}^{k} p_{i+j-1}(\beta_j) < \sigma \\ 0, otherwise \end{cases}$$

where $p_i(\beta_j)$ indicates emission probability from sequence profile $p(X)$ at position i. Compared to the substitution kernel, the score $I_p(\alpha,\beta)$ for two k-mers α and β depends on the *positions* of α and β within the sequences.

The spatial sample kernel [4] is defined over spatial features

$$a_1 \overset{d_1}{\leftrightarrow} a_2, \overset{d_2}{\leftrightarrow}, \cdots, \overset{d_{t-1}}{\longleftrightarrow} a_t$$

representing all possible arrangements of t k-long substrings with the maximum distance between two consecutive substrings in the t-tuple constrained by the distance parameter d. Inclusion of the spatial arrangement information among k-mers results in better performance compared to spectrum / mismatch kernels [4].

4.3 2D representations for protein sequences

For remote homology and protein fold prediction tasks, we use 20-dim BLOSUM amino acid substitution vectors for each amino acid to obtain a $20 \times |X|$ 2D amino acid sequence representation for each 1D amino acid sequence X (i.e., we replace each individual amino acid i in X with a 20-dim vector $b_{i,j}, j = 1 \dots |\Sigma| = 20$, of substitution scores for amino acids i and j). We also use 5-dimensional quantitative descriptors [32] derived from 237 physico-chemical amino acid properties. As we show in the experiments both of these descriptors (20-dim BLOSUM and 5-dim quantitative descriptors) improve predictive accuracy compared to 1D symbolic amino acid representations.

We also use 2D sequence profiles ($20 \times |X|$) and compare with profile kernel approach [2].

4.4 2D kernels for protein sequences

We use state-of-the-art spectrum/mismatch [6] and spatial (SSSK) [4] kernels as our basic 1D similarity kernels (i.e. we use spectrum/mismatch and SSSK similarity

kernels as the row-matching function $I(\cdot,\cdot)$ in Eq. 7). Depending on the choice of the 1D similarity kernel, we refer to corresponding multivariate kernels as 2D-spectrum, 2D-mismatch, and 2D-SSSK multivariate kernels. We use typical settings for kernel parameters and set k=5, m=1 for mismatch kernels, k=1, t=3, d=5 for SSSK kernels. Parameters k, and σ for the profile kernel are set to 5 and 7.5 according to the best found in previous studies [2].

For direct feature quantization, we use uniform quantization of each feature data range into B=32 bins (during testing for values outside of the (f_{min}, f_{max}) range, we use special values of 0 and B+1 for values smaller than f_{min} or larger than f_{max}). For discrete embedding with similarity hashing, we use $E = 8$ bits.

For all tasks, we use kernel SVM [20] as a classifier. For multi-class problems, we use one-vs-rest approach and train a binary classifier for each class.

All experiments are performed on a single $2.8GHz$ CPU. The datasets used in our experiments and the supplementary data/code are available at `http://paul.rutgers.edu/~pkuksa/mvstring.html`.

4.5 Evaluation measures

For multi-class protein fold recognition tasks, the methods are evaluated using 0-1 and top-q balanced error rates as well as F1 scores. Under the top-q error cost function, a classification is considered correct if the rank of the correct label, obtained by sorting all prediction confidences in non-increasing order, is at most q. On the other hand, under the balanced error cost function, the penalty of mis-classifying one sequence is inversely proportional to the number of sequences in the target class (i.e. mis-classifying a sequence from a class with a small number of examples results in a higher penalty compared to that of mis-classifying a sequence from a large, well represented class). Balanced error rates is more indicative of the performance on protein classification tasks since class sizes are unbalanced (vary in size considerably), and the balanced error captures performance on all classes, not just the largest classes.

We evaluate remote protein homology performance using standard Receiver Operating Characteristic (ROC) and ROC50 scores. The ROC50 score is the (normalized) area under the ROC curve computed for up to 50 false positives. With a small number of positive test sequences and a large number of negative test sequences, the ROC50 score is typically more indicative of the prediction accuracy of a homology detection method than the ROC score.

4.6 Remote homology detection

We first compare our proposed *multivariate* string kernel method with a number of state-of-the-art kernel methods for remote homology including spectrum/mismatch kernels [3], [11], spatial sample kernels (SSSK) [4], semi-supervised cluster kernel [27], as well as state-of-the-art profile kernel [2].

To obtain multivariate protein sequence representations, we use rows from the BLOSUM substitution matrix as amino-acid feature vectors, i.e. all of the protein sequences are represented as $20 \times |X|$ multivariate (2D) sequences.

In Table 2, we compare in terms of average ROC and ROC50 univariate kernels (first column), multivariate (2D) kernels using discrete feature quantization (second column), and multivariate (2D) kernels using binary Hamming embedding (third column).

As can be seen from results in Table 2, multivariate string kernel provides effective improvements over other string kernel approaches. For instance, using 2D BLOSUM substitution profiles with spectrum and mismatch kernels significantly improves average ROC50 scores from 27.91 and 41.92 to 43.29 and 49.17, respectively (relative improvements of 50% and 17%), compared to traditional (1D) spectrum/mismatch approaches. Similar improvements observed when using spatial sample kernel (SSSK) (average ROC50 increases from 50.12 using 1D amino acid sequences to 55.54 using 2D BLOSUM representation with SSSK kernel, 11% relative improvement).

We also observe that the multivariate (2D) kernel provides substantial improvements in semi-supervised settings using *semi-supervised* cluster kernel [27] and profile kernel approaches. For example, the multivariate kernel on sequence profiles used by the profile kernel (obtained from non-redundant sequence database (NRDB) [2]) achieves a higher average ROC50 score of 86.27 compared to 81.51 of the profile kernel.

We also note that using direct feature quantization provides more effective improvements compared to discrete embedding with similarity hashing (Table 2). For example, using similarity hashing with the spectrum and SSSK kernels yields smaller improvements compared to direct feature quantization (average ROC50 score of 39.88 vs 43.29, and 54.02 vs 55.54, respectively).

Table 3 shows for each of the kernel methods (spectrum, mismatch, spatial sample, and profile) p-values of the Wilcoxon signed-rank test on the ROC50 scores (54 experiments) against 1D kernels. As can be seen from the table, observed improvements (Table 2) are significant.

Table 5 summarizes classification performance of the discrete (binary) embedding with similarity hashing as a function of the embedding size (E=8,16,32 bits) and the kernel parameters (k=5,8,10).

In Table 4, we also compare with recently proposed spectrum-RBF and mismatch-RBF methods [12] which incorporate *physico-chemical descriptors* with traditional spectrum/mismatch kernels, as well as generative model based methods (Profile HMM from [34], and traditional SVM-Fisher methods [33]), and recent sequence learning methods SEQL [28].

We note the our multivariate similarity kernel using

TABLE 2: Classification performance (mean ROC50) on protein remote homology detection (54 experiments)

Method	Univariate (1D)		Multivariate (2D) DFQ		Multivariate (2D) Sim. hashing	
	Mean ROC	Mean ROC50	Mean ROC	Mean ROC50	Mean ROC	Mean ROC50
Spectrum [11]	78.13	27.91	88.53	43.29	87.23	39.88
Mismatch-(k=5,m=1) [3]	87.75	41.92	91.03	49.17	90.94	49.38
Spatial sample (SSSK) [4]	90.21	50.12	92.17	55.54	91.68	54.02
Profile kernel-(k=5,σ=7.5) (NRDB) [2]	97.34	81.51	98.45	86.27		

TABLE 3: Statistical significance of the differences between 1D (amino acid sequence) kernels and the proposed multivariate kernels (remote homology detection). Multivariate similarity kernels perform better than the traditionally used 1D (univariate) kernels

Method	p-value	effect size	relative improvement (2D vs 1D), %
Spectrum kernel	2.5e-6	0.58	50.6
Mismatch kernel	6.5e-3	0.26	17.3
Spatial sample kernel (SSSK)	2.7e-5	0.20	10.8
Profile kernel	2.9e-4	0.26	6.3

TABLE 4: Protein remote homology detection. Comparison with generative and other state-of-the-art methods

Method	Mean ROC	Mean ROC5
2D-SSSK	92.17	**55.54**
Spectrum-RBF [12]†	-	42.1
Mismatch-RBF [12]†	-	43.6
SVM-Fisher [33]	75.66	31.90
Profile HMM [34]	88.33	49.16
SEQL [28]	92.20	52.17

† results from [12]

TABLE 5: Remote homology prediction. Classification performance (mean ROC50) as a function of the embedding size E and the kernel parameters

Embedding size E	k=5	k=8	k=10
8	38.53	37.53	37.96
16	39.59	38.54	36.41
32	39.88	39.72	38.43

only BLOSUM substitution scores achieves higher average ROC50 scores (Table 4) compared to computationally more expensive spectrum-RBF/mismatch-RBF approaches [12] which exploit richer *physico-chemical* descriptors (BLOSUM, AAindex descriptors, etc).

Table 6 shows results on the second protein remote homology benchmark (SCOP 1.53) and compares the performance of the proposed 2D kernels (2D-Mismatch and 2D-SSSK) with the univariate string kernels (mismatch and SSSK), as well as recently developed SVM prediction methods that use physico-chemical amino-acid descriptors (SVM-PCD [14], SVM-RQA [13]. As can be seen from the table, multivariate (2D) kernels display the highest ROC50 performance on this benchmark.

4.7 Protein fold detection

We compare performance on SCOP 1.73 fold benchmark in Table 7. Similar to the observed improvements on protein remote homology detection (Table 2), on this more challenging fold detection task, multivariate (2D) kernels (2D-Mismatch, 2D-SSSK) provide effective improvement over corresponding univariate mismatch and SSSK kernels (e.g., we observe increase in average ROC50 scores from 24.34 to 27.64 using 2D-SSSK kernel as opposed to the univariate SSSK kernel).

4.8 Multi-class protein fold prediction

For multi-class protein fold prediction (Table 8, Ding&Dubchack dataset, 27-folds), using the multivariate string kernel with BLOSUM profiles ($20 \times |X|$) we observe substantial improvements over 1D mismatch kernel, e.g., balanced error rate improves from 53.2% to 48.5% for mismatch-(k=5,m=1) kernel (9% relative improvement). We also note that obtained error rates compare well with the error rates of computationally more expensive substitution kernel [6] which also uses BLOSUM substitution scores to measure similarity between k-mers.

On a challenging remote fold prediction dataset [7] (results in Table 9), we observe similar improvements in ranking quality when using the multivariate kernel with BLOSUM profiles over corresponding string kernel methods which use 1D amino acid sequences. For instance, 28.92% top-5 error rate of the cluster kernel with BLOSUM profile compares well with 35.28% error rate of the state-of-the-art profile kernel.

4.9 MHC binding prediction

We test MHC binding prediction performance on three MHC alleles (A*2301,B*5801,A*0201) with small (104 peptides), moderate (988 peptides), and large (3089 peptides) number of binding peptides. The classification performance (binding vs non-binding) is evaluated using average ROC scores over 5-fold cross-validation (cross-validation folds are the same as in [30]). Table 10 compares performance of the traditional spectrum and weighted degree (WD) kernels [35] which use position-specific matching. As can be seen from the results, using

TABLE 6: Comparison with univariate string kernel and SVM physico-chemical properties-based predictors on SCOP 1.53 benchmark

Method	ROC	ROC50
Mismatch-(5,1)	87.25	40.39
2D-Mismatch-(5,1)	89.73	48.20
SSSK	89.80	51.79
2D-SSSK	90.45	55.22
SVM-RQA [13]	91.2	44.1
SVM-PCD [14]	90.2	-

TABLE 7: Protein remote fold detection performance on SCOP 1.73 fold benchmark

Method	ROC	ROC50
Mismatch-(5,1)	76.74	20.13
SSSK	78.96	24.34
2D-Mismatch	78.59	24.88
2D-SSSK	80.54	27.64

TABLE 8: Multi-class protein fold prediction [29] (27-class)

Method	Error, %	Balanced error, %	F1
Baseline 1: Mismatch-(k=5,m=1)	51.17	53.22	61.68
Baseline 2: Substitution kernel [6]	45.43	**48.02**	53.54
2D-Spectrum	**43.86**	48.49	**63.18**

TABLE 9: Classification performance on fold prediction (multi-class) [7]

Method	Error	Top 5 Error	Balanced Error	Top 5 Balanced Error	F1	Top 5 F1
Baseline 1: PSI-BLAST [7]	64.80	51.80	70.30	54.30	-	-
Baseline 2: Substitution kernel [6]	51.95	27.04	66.17	36.72	34.49	66.27
Baseline 3: Profile (5,7.5) (Swiss-prot) [2]	49.35	20.36	76.67	35.28	26.05	68.09
Spatial sample kernel (SSSK) [4]	48.7	25.08	73.04	44.05	30.57	62.37
2D-SSSK	45.77	20.19	68.99	35.90	33.41	68.09
Mismatch-(k=5,m=1) [3]	53.75	29.15	82.75	52.40	16.92	56.67
2D-Mismatch-(k=5,m=1)	47.88	21.17	73.09	34.31	29.16	70.09
Semi-supervised Cluster kernel (Swiss-Prot) [27]	48.86	19.54	72.88	34.06	26.59	70.07
2D-Semi-supervised Cluster kernel (Swiss-Prot)	48.86	18.40	74.87	28.92	27.06	74.24

multivariate (2D) representations improves performance over 1D representations (spectrum and weighted degree). Observed improvements are more significant for small and moderate MHC datasets (A*2301,B*5801), e.g., using multivariate BLOSUM representation achieves higher average ROC score of 83.28 compared to 78.63 ROC of 1D spectrum kernel.

4.10 Running time

In Table 11, we compare the running time for the proposed multivariate string kernel and traditional univariate string kernel methods. We note that for mismatch-(k,m) kernel computation (protein remote homology data) we use linear time sufficient-statistic based algorithms from [10]. As can be seen from results, using multivariate similarity kernels gives similar performance in running times compared to traditional 1D (univariate) kernels while displaying better classification performance (e.g., Table 2).

5 Conclusions

We presented new *multivariate* string kernel methods for biological sequence classification that exploit richer multivariate feature sequence representations (biological sequence profiles, and sequences of amino acid descriptors, in particular). The proposed approach directly exploits these *multivariate* feature sequences (2D) to improve sequence classification. On three protein sequence classification tasks this shows a significant 15-20% improvement compared to state-of-the-art sequence classification methods.

References

[1] J. Cheng and P. Baldi, "A machine learning information retrieval approach to protein fold recognition," *Bioinformatics*, vol. 22, no. 12, pp. 1456–1463, June 2006. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/22/12/1456

[2] R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. S. Leslie, "Profile-based string kernels for remote homology detection and motif extraction." in *CSB*, 2004, pp. 152–160.

[3] C. S. Leslie, E. Eskin, J. Weston, and W. S. Noble, "Mismatch string kernels for SVM protein classification." in *NIPS*, 2002, pp. 1417–1424.

[4] P. P. Kuksa and V. Pavlovic, "Spatial representation for efficient sequence classification," in *ICPR*, 2010.

[5] S. Sonnenburg, G. Rätsch, and B. Schölkopf, "Large scale genomic sequence SVM classifiers," in *ICML '05*, New York, NY, USA, 2005, pp. 848–855.

[6] C. Leslie and R. Kuang, "Fast string kernels using inexact matching for protein sequences," *J. Mach. Learn. Res.*, vol. 5, pp. 1435–1455, 2004. [Online]. Available: http://jmlr.csail.mit.edu/papers/volume5/leslie04a/leslie04a.pdf

[7] I. Melvin, E. Ie, J. Weston, W. S. Noble, and C. Leslie, "Multi-class protein classification using adaptive codes," *J. Mach. Learn. Res.*, vol. 8, pp. 1557–1581, 2007.

TABLE 10: Classification performance (ROC scores) on MHC-peptide binding prediction

method	A*2301	B*5801	A*0201
Baseline 1: WD [12]	73.07	93.14	94.85
Baseline 2: Spectrum	78.63	92.05	95.05
2D-Spectrum	83.28	94.11	95.14

TABLE 11: Running time for the kernel computations

Method	Running time (s), kernel matrix computation
Mismatch(5,1)	13.1
Mismatch(5,2)	76.5
2D-Spectrum (BLOSUM)	91.5
2D-Mismatch (BLOSUM)	245

[8] P. Kuksa and V. Pavlovic, "Efficient alignment-free dna barcode analytics," *BMC Bioinformatics*, vol. 10, no. Suppl 14, p. S9, 2009, impact factor: 3.78. [Online]. Available: http://www.biomedcentral.com/1471-2105/10/S14/S9

[9] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. New York, NY, USA: Cambridge University Press, 2004.

[10] P. Kuksa, P.-H. Huang, and V. Pavlovic, "Scalable algorithms for string kernels with inexact matching," in *NIPS*, 2008.

[11] C. S. Leslie, E. Eskin, and W. S. Noble, "The spectrum kernel: A string kernel for SVM protein classification." in *Pacific Symposium on Biocomputing*, 2002, pp. 566–575.

[12] N. Toussaint, C. Widmer, O. Kohlbacher, and G. Ratsch, "Exploiting physico-chemical properties in string kernels," *BMC Bioinformatics*, vol. 11, no. Suppl 8, p. S7, 2010. [Online]. Available: http://www.biomedcentral.com/1471-2105/11/S8/S7

[13] Y. Yang, E. Tantoso, and K.-B. Li, "Remote protein homology detection using recurrence quantification analysis and amino acid physicochemical properties," *Journal of Theoretical Biology*, vol. 252, no. 1, pp. 145 – 154, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022519308000453

[14] B.-J. Webb-Robertson, K. Ratuiste, and C. Oehmen, "Physicochemical property distributions for accurate and rapid pairwise protein homology detection," *BMC Bioinformatics*, vol. 11, no. 1, p. 145, 2010. [Online]. Available: http://www.biomedcentral.com/1471-2105/11/145

[15] R. D. King, A. Karwath, A. Clare, and L. Dehaspe, "The utility of different representations of protein sequence for predicting functional class," *Bioinformatics*, vol. 17, no. 5, pp. 445–454, 2001. [Online]. Available: http://bioinformatics.oxfordjournals.org/content/17/5/445.abstract

[16] C. Z. Cai, L. Y. Han, Z. L. Ji, X. Chen, and Y. Z. Chen, "Svm-prot: web-based support vector machine software for functional classification of a protein from its primary sequence," *Nucleic Acids Research*, pp. 3692–3697, 2003.

[17] C. S. Ong and A. Zien, "An automated combination of kernels for predicting protein subcellular localization," in *Proceedings of the 8th international workshop on Algorithms in Bioinformatics*, ser. WABI '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 186–197. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-87361-7_16

[18] T. Hertz and C. Yanover, "PepDist: A new framework for protein-peptide binding prediction based on learning peptide distance functions," *BMC Bioinformatics*, vol. 7, no. Suppl 1, pp. S3+, 2006. [Online]. Available: http://dx.doi.org/10.1186/1471-2105-7-S1-S3

[19] N. Pfeifer and O. Kohlbacher, "Multiple instance learning allows mhc class ii epitope predictions across alleles," in *Proceedings of the 8th international workshop on Algorithms in Bioinformatics*, ser. WABI '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 210–221. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-87361-7_18

[20] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, September 1998. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0471030031

[21] C. Cortes, P. Haffner, and M. Mohri, "Rational kernels: Theory and algorithms," *Journal of Machine Learning Research*, vol. 5, pp. 1035–1062, 2004.

[22] S. V. N. Vishwanathan and A. Smola, "Fast kernels for string and tree matching," in *NIPS*, 2002.

[23] M. Gribskov, A. McLachlan, and D. Eisenberg, "Profile analysis: detection of distantly related proteins," *Proceedings of the National Academy of Sciences*, vol. 84, pp. 4355–4358, 1987.

[24] L. J. Jensen, R. Gupta, N. Blom, D. Devos, J. Tamames, C. Kesmir, H. Nielsen, H. H. Strfeldt, K. Rapacki, C. Workman, C. A. F. Andersen, S. Knudsen, A. Krogh, A. Valencia, S. Brunak, and B. Cnb-csic, "Prediction of human protein function from post-translational modifications and localization features," *J Mol Biol*, vol. 319, pp. 1257–1265, 2002.

[25] I. Dubchak, I. Muchnik, S. R. Holbrook, and S. H. Kim, "Prediction of protein folding class using global description of amino acid sequence," *Proceedings of the National Academy of Sciences*, vol. 92, no. 19, pp. 8700–8704, 1995. [Online]. Available: http://www.pnas.org/content/92/19/8700.abstract

[26] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., 2009, pp. 1753–1760.

[27] J. Weston, C. Leslie, E. Ie, D. Zhou, A. Elisseeff, and W. S. Noble, "Semi-supervised protein classification using cluster kernels," *Bioinformatics*, vol. 21, no. 15, pp. 3241–3247, 2005. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/21/15/3241

[28] G. Ifrim and C. Wiuf, "Bounded coordinate-descent for biological sequence classification in high dimensional predictor space," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '11. New York, NY, USA: ACM, 2011, pp. 708–716. [Online]. Available: http://doi.acm.org/10.1145/2020408.2020519

[29] C. H. Ding and I. Dubchak, "Multi-class protein fold recognition using support vector machines and neural networks," *Bioinformatics*, vol. 17, no. 4, pp. 349–358, 2001. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/17/4/349

[30] B. Peters, H.-H. Bui, S. Frankild, M. Nielsen, C. Lundegaard, E. Kostem, D. Basch, K. Lamberth, M. Harndahl, W. Fleri, S. S. Wilson, J. Sidney, O. Lund, S. Buus, and A. Sette, "A community resource benchmarking predictions of peptide binding to mhc-i molecules," *PLoS Comput Biol*, vol. 2, no. 6, p. e65, 06 2006. [Online]. Available: http://dx.plos.org/10.1371%2Fjournal.pcbi.0020065

[31] L. Lo Conte, B. Ailey, T. Hubbard, S. Brenner, A. Murzin, and C. Chothia, "SCOP: a structural classification of proteins database," *Nucleic Acids Res.*, vol. 28, pp. 257–259, 2000.

[32] M. S. Venkatarajan and W. Braun, "New quantitative descriptors of amino acids based on multidimensional scaling of a large number of physical–chemical properties," *Journal of Molecular Modeling*, vol. 7, pp. 445–453, 2001, 10.1007/s00894-001-0058-5. [Online]. Available: http://dx.doi.org/10.1007/s00894-001-0058-5

[33] T. Jaakkola, M. Diekhans, and D. Haussler, "Using the Fisher kernel method to detect remote protein homologies," in *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, 1999, pp. 149–158.

[34] P. Huang and V. Pavlovic, "Protein homology detection with biologically inspired features and interpretable statistical models," *Int. J. Data Min. Bioinformatics*, vol. 2, no. 2, pp. 157–175, Jun. 2008. [Online]. Available: http://dx.doi.org/10.1504/IJDMB.2008.019096

[35] G. Raetsch and S. Sonnenburg, "Accurate splice site detection for caenorhabditis elegans," *MIT Press series on Computational Molecular Biology*, pp. 277–298, 2004.

Pavel Kuksa received his bachelor's degree in Computer Engineering with honors in 2002 and master's degree with honors in Information and Computer Sciences in 2005, both from Bauman Moscow State Technical University. He then completed his Ph.D. in the Department of Computer Science at Rutgers University in 2011. His research has always been focused on advancing the state-of-the-art both in theory and practice of machine learning algorithms, algorithms for sequence and time series analysis, bio-informatics, natural language processing, and large-scale data analysis.