VECTOR QUANTIZATION KERNELS FOR THE CLASSIFICATION OF PROTEIN SEQUENCES AND STRUCTURES

WYATT T. CLARK AND PREDRAG RADIVOJAC*

Department of Computer Science and Informatics, Indiana University
Bloomington, Indiana 47405, U.S.A.
**E-mail: predrag@indiana.edu*

We propose a new kernel-based method for the classification of protein sequences and structures. We first represent each protein as a set of time series data using several structural, physicochemical, and predicted properties such as a sequence of consecutive dihedral angles, hydrophobicity indices, or predictions of disordered regions. A kernel function is then computed for pairs of proteins, exploiting the principles of vector quantization and subsequently used with support vector machines for protein classification. Although our method requires a significant pre-processing step, it is fast in the training and prediction stages owing to the linear complexity of kernel computation with the length of protein sequences. We evaluate our approach on two protein classification tasks involving the prediction of SCOP structural classes and catalytic activity according to the Gene Ontology. We provide evidence that the method is competitive when compared to string kernels, and useful for a range of protein classification tasks. Furthermore, the applicability of our approach extends beyond computational biology to any classification of time series data.

Keywords: Protein classification, protein structure, protein function, kernels, vector quantization, support vector machines.

1. Introduction

The wealth and diversity of experimental data in the life sciences has strongly influenced the development of classification methods for biological macromolecules. Over the past couple of decades the scope and sophistication of these methods has significantly increased, leading to the adoption of classification schemes that are designed to integrate diverse types of biological data (e.g. sequence, structure, interaction networks, text), enable principled incorporation of domain knowledge, and rigorously deal with data of varying degrees of quality.[1,2]

Among the various classification strategies, kernel-based methods[3,4] have recently been introduced in a range of contexts such as the prediction of remote homology,[5] protein structure[6] and function,[7,8] protein-protein interactions,[9] gene-disease associations,[10] the activity of chemical compounds,[11] etc. Although some kernel methods have been developed to predict properties of individual residues,[12] most of these approaches have been used at the level of entire proteins. For example, several string kernels were introduced to provide inferences regarding remote homology from amino acid sequences.[5,13–15] Similarly, graph kernels have gained significant attention owing to the fact that a variety of biological data can be modeled through graphs.[16] A number of approaches have also considered integrating kernels built on different types of data.[17,18]

Kernels can be roughly described as symmetric positive semi-definite similarity functions that operate on pairs of objects from some input space.[19] Their mathematical properties guarantee the existence of a Hilbert space, potentially of infinite dimensionality, such that the value of the kernel function can be computed as the inner product of the images of

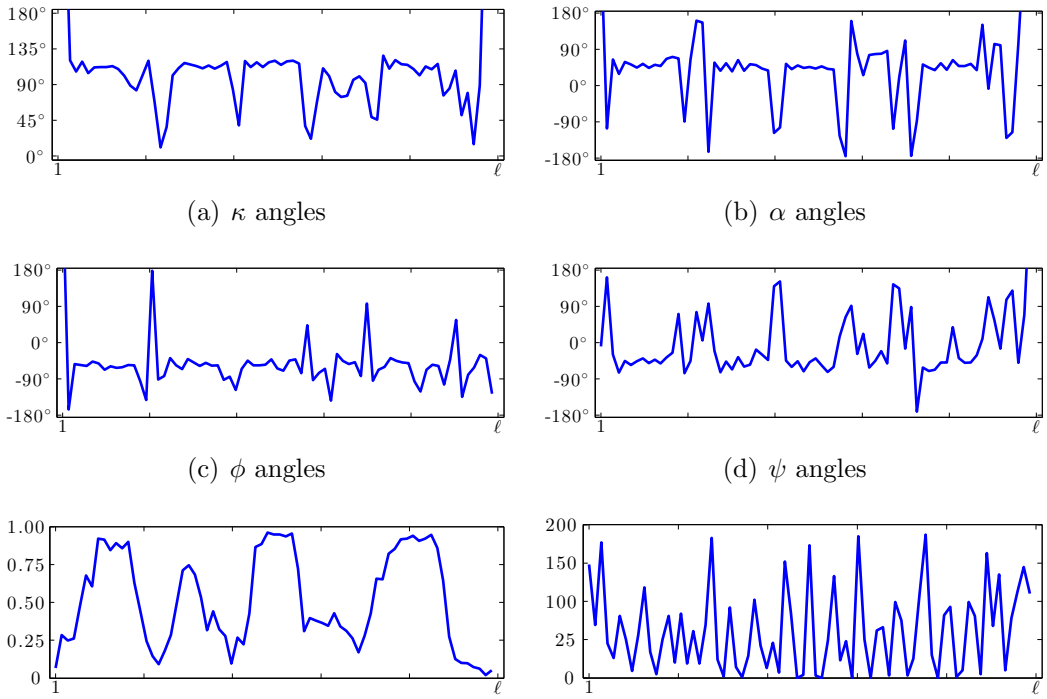


Fig. 1. Time series representation of various protein properties for the $\ell = 73$ amino acid long DNA helicase RuvA subunit d1ixra1 from *Thermus thermophilus* (PDB ID 1ixr).

the input objects. When coupled with learning algorithms such as support vector machines, kernel functions also guarantee a globally optimal solution to the optimization problem.[19] Although most kernel-based approaches are in practice formed by vectorizing input objects, thereby not fully exploiting their theoretical potential, they still enable a practitioner to incorporate domain knowledge into modeling the relationship between objects, rather than simply encoding properties of the objects into a potentially high-dimensional vector space and providing them to a standard classifier.

In this work we focus on kernel-based strategies and develop novel methodology for the nonalignment-based classification of proteins into distinct categories. In contrast to most previously implemented kernel approaches, we represent a protein's sequence or structure, if available, as a set of time series properties (we consider a time series to be an ordered sequence of real-valued numbers[20]). One such time series representation of a DNA helicase subunit from *Thermus thermophilus* is shown in Figure 1, where six different types of properties have been generated based on the protein's sequence and structure. Given the time series data, we utilize ideas from vector quantization (VQ), initially developed for lossy signal compression,[21] to define a kernel function between pairs of protein sequences that we use for classification. We extensively evaluated methods on two distinct and relevant problems: (*i*) the classification of protein structures into structural classes and (*ii*) the prediction of protein function from amino acid sequence. Our experiments provide evidence that the new kernels are a viable approach in various practical scenarios.

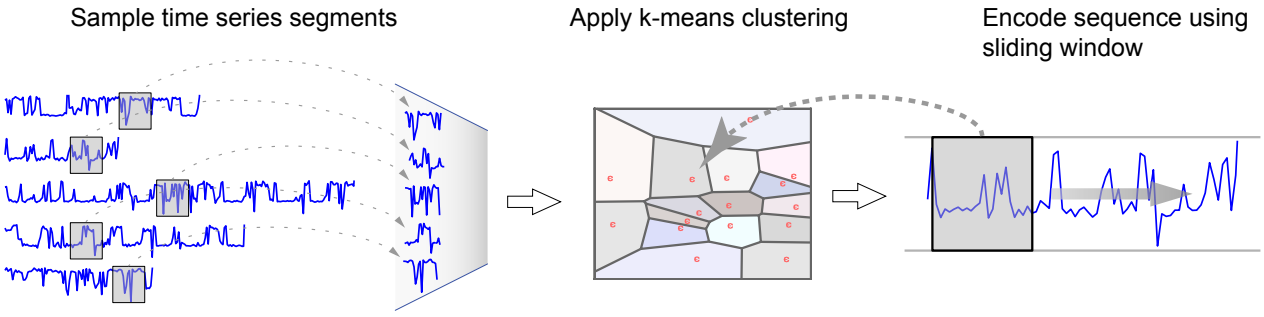


Fig. 2. A schematic representation of using VQ to encode a sequence represented as a property vector. First individual time series property vectors are broken up into n length segments as shown on the left. These sub-sampled vectors from a database of sequences are then used to create a clustering in n-dimensional space as shown in the center. Finally, an original property vector is encoded using the derived set of centroids by counting the number of overlapping sub-segments which are the closest to each centroid.

2. Methods

Let $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, ...\}$ be a universe of protein sequences, where each $\mathbf{s} \in \mathcal{S}$ is a string of symbols from an alphabet of amino acids $\mathcal{A} = \{$A, C, D, ..., Y$\}$. Let also $\mathcal{S}_L \subset \mathcal{S}$ be a set of labeled sequences, e.g. those with known structural class or function, that is provided as training data. The objective is to use an inductive supervised framework to probabilistically annotate the remaining sequences, i.e. those from the unlabeled set $\mathcal{S}_U = \mathcal{S} - \mathcal{S}_L$.

To map protein sequences into a real-valued vector representation, let $\mathbf{s} = s_1 s_2 \dots s_\ell$ be a length-ℓ protein sequence in $\mathcal{S}$ and $\mathbf{p} = (p_1, p_2, \dots, p_\ell)$ some property vector defined by any particular mapping from $\mathcal{A}^\ell$ to $\mathbb{R}^\ell$. For example, $\mathbf{p}$ may be provided as a vector of hydrophobicity indices corresponding to amino acids in $\mathbf{s}$. Alternatively, it can be represented as predicted helical propensities as outputted by some predictor of secondary structure. For those sequences with available structures, $\mathbf{p}$ may correspond to a sequence of dihedral angles calculated from the protein structure model of $\mathbf{s}$.

Consider now a single property vector $\mathbf{p}$, such as a hydrophobicity profile, corresponding to a particular sequence $\mathbf{s} \in \mathcal{S}$. We decompose $\mathbf{p}$ into n-dimensional overlapping sub-vectors $\mathbf{p}_{[1,n]}$, $\mathbf{p}_{[2,n+1]}$, ..., $\mathbf{p}_{[\ell-n+1,\ell]}$, where $\mathbf{p}_{[i,i+j]} = (p_i, p_{i+1}, ..., p_{i+j})$, and $n \ll \ell$ is a small integer. For example, $\mathbf{p}_{[1,n]}$ corresponds to the first n elements of $\mathbf{p}$. For a property vector (amino acid sequence) of length ℓ, there are $\ell - n + 1$ length-n sub-vectors.

As described in Figure 2, given a set of length-n property sub-vectors $\mathcal{P}$ derived from the sequence universe $\mathcal{S}$, we generate a partition of $\mathbb{R}^n$ into m regions $\mathcal{R} = \{R_1, R_2, \dots, R_m\}$. These regions are represented by a set of n-dimensional vectors, or centroids, $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_m\}$. Each region R_i represents a Voronoi region such that

$$R_i = \{\mathbf{x} : d(\mathbf{x}, \mathbf{c}_i) \leq d(\mathbf{x}, \mathbf{c}_j), i \neq j\},$$

where $d(\mathbf{x}, \mathbf{c})$ is the Euclidean distance between vector $\mathbf{x}$ and centroid $\mathbf{c}$. We determine $\mathcal{C}$ using k-means clustering, where the initial set of clusters is generated by the splitting method.[22] We chose to use k-means clustering as opposed to simply creating a lattice in n-dimensional space,[23] because sampled property vectors do not fill the space evenly, but instead cluster

around evolutionarily conserved or sterically preferred regions.

2.1. *Property kernels*

Variable length property vectors can be transformed into vectors of length m using the partition of $\mathbb{R}^n$ defined by $\mathcal{R}$. Specifically, a property vector $\mathbf{p}$ is mapped into a vector of length m as

$$\mathbf{x} = (\varphi_1(\mathbf{p}), \varphi_2(\mathbf{p}), \ldots, \varphi_m(\mathbf{p}))\,,$$

where $\varphi_i(\mathbf{p})$ is the number of n-dimensional vectors $\mathbf{p}_{[.]}$ in $\mathbf{p}$ that belong to region R_i. Given two property vectors $\mathbf{p}$ and $\mathbf{q}$ and their respective count vectors $\mathbf{x}$ and $\mathbf{y}$, a *vector quantization property kernel* function is defined as

$$k(\mathbf{p}, \mathbf{q}) = \mathbf{x}^T\mathbf{y},$$

where T is the transpose operator. Note that in this notation each count vector is assumed to be a column vector, i.e. $(a, b, c) = [a\ b\ c]^T$, as in Ref. 24. Since the function $k(\mathbf{p}, \mathbf{q})$ is defined as an inner product between two count vectors, it is a kernel function.[19]

Given a set of property kernels $\{k_i(\mathbf{x}, \mathbf{y})\}$, we construct the composite property kernel as a linear combination

$$k(\mathbf{x}, \mathbf{y}) = \sum_i k_i(\mathbf{x}, \mathbf{y}),$$

where before and after combining, each kernel is normalized using

$$k(\mathbf{x}, \mathbf{y}) \leftarrow \frac{k(\mathbf{x}, \mathbf{y})}{\sqrt{k(\mathbf{x}, \mathbf{x})k(\mathbf{y}, \mathbf{y})}}.$$

It is important to mention that both the inner product kernel formulation and the composite kernel based on a linear combination were selected for their simplicity. Functions such as the Jaccard similarity coefficient and the Gaussian kernel (which introduces a parameter into kernel selection) sometimes provide performance improvements to an inner product definition. Similarly, kernels can be combined using a product or hyperkernel formulations; however, recent evidence suggests that more sophisticated schemes typically result in only minor improvements over linear combinations.[25]

2.2. *Computational complexity*

The computation of a count vector can be accomplished in $O(\ell mn)$ time if each n-dimensional vector from $\mathbf{p}$ is compared with all centroids in $\mathcal{C}$. Approximation algorithms are available through a decision tree-like organization of the centroids. In such a case, only $\log m$ distance calculations are needed resulting in $O(\ell n \log m)$ time;[22] however, there is no guarantee that the closest centroid will be found. The memory requirements include $O(mn)$ space for storing $\mathcal{C}$.

2.3. *Spectrum kernel*

We compare the property kernels to a string kernel approach, as described in Ref. 5, for a wide range of word sizes ($n \in \{1 \dots 10\}$). For a given word size n, a sparse 20^n-length vector was created for a protein sequence, where each dimension represented the number of times a potential substring of length n that could be generated using the 20 amino acid alphabet occurred. An ℓ-length sequence, $\mathbf{s}$, contains $\ell - n + 1$ such overlapping strings.

3. Data and experiments

In the first experiment, prediction was performed as a one-versus-all classification at the SCOP class level for single domain proteins categorized as α, β, $\alpha + \beta$, or α/β.[26] We utilized Astral 1.75A (40%) to ensure that redundancy in the data set did not lead to inflated assessment of performance. Table 1 summarizes the positive and negative data points used for each category of SCOP.

In the second experiment we attempted to distinguish enzymes, or those proteins annotated with the term "catalytic activity" and its subtypes, from all other proteins. Gene Ontology[27] (GO) annotations were obtained from the April 2012 release of Swiss-Prot[28] in conjunction with the May 4, 2012 version of GO. Only annotations supported by evidence codes EXP, IDA, IPI, IMP, IGI, IEP, TAS and IC were used. This resulted in a total of 24,882 proteins with experimentally verified annotations, 9,506 of which were annotated with the term "catalytic activity" and 18,936 of which represented putatively negative data points.

Table 1. Summary of data used for SCOP classification documenting the number of positives and negatives used for the classification of protein structures as α, β, $\alpha + \beta$, or α/β.

SCOP class	positives	negatives
all α	1,901	7,486
all β	2,175	7,212
$\alpha + \beta$	2,665	6,722
α/β	2,646	6,741

We tested a range of combinations of values for n (window size), and m (number of centroids) for each property. For values of n, we tested $n \in \{2^i : i = 1 \dots 5\}$. Similarly, for the number of centroids, m, we tested $m \in \{2^i : i = 4, 6, 8, 10, 12\}$. For each property type and all values of m and n, we performed k-means clustering using 10^6 randomly sampled vectors from all sequences in $\mathcal{S}$. SVM^{light} with the default value for the capacity parameter was used as a prediction engine in all experiments.[29] In each experiment, the total costs of misclassification for positive and negative examples were equal.

3.1. *Mapping proteins into property vectors*

Several structure-based properties were generated by converting the atomic 3D coordinates into backbone angles. The usefulness of representing a structure in this manner is that backbone angles are invariant to the translation and rotation of the original 3D coordinates. Four types of backbone angles were utilized: α, κ, ϕ, and ψ. All angles were obtained using DSSP.[30] In addition to generating dihedral angles, DSSP also outputs solvent accessibility values which we used as the fifth structure-based property.

We also generated seven sequence-based properties for both the task of categorizing structures and predicting function. These features were generated in order to represent biologically

Table 2. Optimal performance, according to AUC, of each property-based feature when predicting SCOP folds. For each property feature the combination of m and n values that obtained the highest AUC for an individual SCOP category are reported. The last block of columns shows the weighted AUC, $AUC^{\mathbf{w}}$, obtained across all SCOP categories. Results when combining all sequence- and structure-based features are shown in the bottom section of the table. Because different combinations of m and n were used when combining properties, these values are not shown for the Sequence + Structure and Sequence + Structure + String kernel combination of features.

Property\Category	All α			All β			α/β			$\alpha+\beta$			$AUC^{\mathbf{w}}$		
	m	n	AUC	m	n	AUC	m	n	AUC	m	n	AUC	m	n	$AUC^{\mathbf{w}}$
α angles	4,096	8	0.994	4,096	8	0.982	4,096	16	0.967	4,096	32	0.829	–	–	0.930
κ angles	4,096	16	0.995	4,096	16	0.986	4,096	16	0.978	4,096	32	0.903	–	–	0.961
ϕ angles	4,096	16	0.990	4,096	16	0.975	4,096	16	0.964	4,096	32	0.809	–	–	0.920
ψ angles	4,096	8	0.991	4,096	16	0.981	4,096	16	0.970	4,096	32	0.850	–	–	0.938
Solvent accessibility	4,096	8	0.960	4,096	8	0.951	4,096	32	0.914	4,096	32	0.710	–	–	0.868
B-factor predictions	256	8	0.790	256	8	0.709	64	8	0.809	16	4	0.587	–	–	0.717
Helix predictions	16	4	0.868	16	8	0.871	64	16	0.842	64	32	0.637	–	–	0.788
Hydrophobicity indices	256	4	0.711	1,024	4	0.728	64	2	0.834	64	2	0.585	–	–	0.707
Loop predictions	256	16	0.872	64	8	0.848	16	32	0.821	1,024	32	0.607	–	–	0.771
PDB disorder predictions	1,024	8	0.842	64	4	0.849	64	4	0.819	256	32	0.591	–	–	0.764
Sheet predictions	64	2	0.904	64	8	0.853	256	16	0.836	64	32	0.641	–	–	0.787
VSL2B disorder predictions	16	2	0.699	64	32	0.628	64	8	0.812	64	2	0.589	–	–	0.682
String kernel	–	2	0.863	–	3	0.878	–	4	0.860	–	5	0.634	–	–	0.794
Sequence	64	16	0.915	64	16	0.876	64	16	0.880	64	16	0.621	64	16	0.813
Structure	4,096	32	0.992	4,096	32	0.983	4,096	32	0.979	4,096	32	0.903	4,096	32	0.961
Sequence + Structure	–	–	0.989	–	–	0.967	–	–	0.970	–	–	0.851	–	–	0.939
Sequence + Structure + String kernel	–	–	0.989	–	–	0.967	–	–	0.970	–	–	0.851	–	–	0.939

relevant properties associated with a region of a protein sequence: (*i*) hydrophobicity, calculated using the the Kyte-Doolittle scale[31] in a sliding window of length $w = 11$; (*ii*) flexibility, calculated as predicted B-factors using our previous model;[32] secondary structure predictions of (*iii*) helix, (*iv*) sheet and (*v*) loop propensities using our in-house predictor; and intrinsic disorder, (*vi*) using the previously published VSL2B model[33] as well as (*vii*) predictions from the same in-house predictor used for secondary structures.

3.2. *Performance evaluation*

We performed 10-fold cross-validation in all experiments. For each binary classification task we calculated the area under the Receiver Operating Characteristic (ROC) curve (*AUC*). While we evaluated each feature type for a combination of window size and number of clusters on each classification task separately, we also desired to obtain a single value that could be used to benchmark each combination of parameters on all evaluated SCOP classes. To do this we used a weighted average of *AUC* values across multiple one-versus-all classification tasks, where the weight for each task was calculated using the ratio of structures in the given category and the total number of structures. We refer to this performance measure as $AUC^{\mathbf{w}}$.

We also calculated the signal-to-noise ratio (*SNR*) obtained when encoding and decoding property-based representations of proteins using vector quantization. Given an original property vector $\mathbf{p}$ and the reconstructed version of this vector $\hat{\mathbf{p}}$, the signal-to-noise ratio was calculated using the logarithmic decibel scale as

$$SNR_{dB}(\mathbf{p}, \hat{\mathbf{p}}) = \log_{10} \frac{\sum_{i=1}^{\ell} p_i^2}{\sum_{i=1}^{\ell} (p_i - \hat{p}_i)^2}.$$

On this scale one decibel signifies that the noise (or sum of squared differences between the original and reconstructed signals) represents $^1/_{10}$-th of the signal.

4. Results

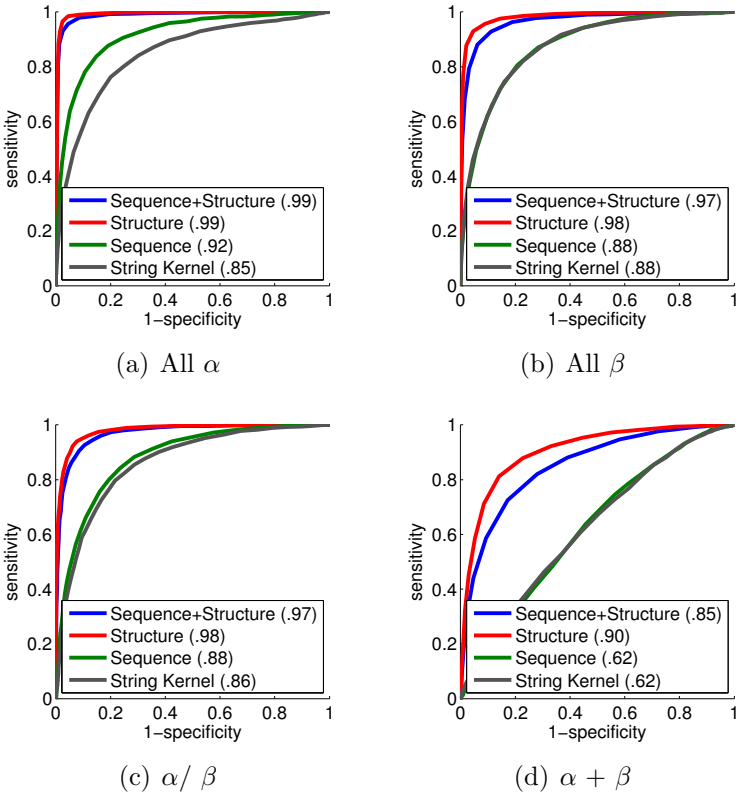
4.1. *Prediction of structural categories*

Table 2 shows the performance of each property kernel when predicting SCOP classes. Among structure-based properties we found that κ angles had the best performance, both for individual SCOP classes and in terms of its $AUC^{\mathbf{w}}$ (0.961). Solvent accessibility values performed the worst out of structure-based properties, obtaining the lowest *AUC* for all SCOP classes as well as lowest $AUC^{\mathbf{w}}$ (0.868). All structure-based properties outperformed sequence-based properties.

Among sequence-based properties, predicted secondary structures performed the best, especially predictions that a residue is in a helix ($AUC^{\mathbf{w}} = 0.788$), a sheet ($AUC^{\mathbf{w}} = 0.787$) or a loop ($AUC^{\mathbf{w}} = 0.771$). Calculated hydrophobicity and VSL2B-based predictions of disorder propensity performed the worst ($AUC^{\mathbf{w}} = 0.707$ and $AUC^{\mathbf{w}} = 0.682$, respectively). Interestingly, the predictor of disordered residues developed from PDB performed considerably better than VSL2B ($AUC^{\mathbf{w}} = 0.764$ vs. $AUC^{\mathbf{w}} = 0.682$), an outcome that may be due to the differences in training samples between the two models.

In order to test the predictive ability of integrating multiple properties, we implemented a linear combination of individual property kernels (Table 2, Figure 3). To reduce the computational complexity of this task we only combined properties utilizing m and n values that achieved the highest $AUC^{\mathbf{w}}$ for each property type.

We observed an improvement of about three percentage points when combining sequence-based properties, achieving an $AUC^{\mathbf{w}}$ of 0.813 compared to the best performance of an individual sequence based property of 0.788 (helix predictions). The combined kernel for structure-based properties saw no improvement over the best performing individual model ($AUC^{\mathbf{w}} = 0.961$ for both the combined kernel and κ angles), and actually exhibited lower performance when both sequence and structure-based properties were combined ($AUC^{\mathbf{w}} = 0.939$).



(a) All α (b) All β

(c) $\alpha/\ \beta$ (d) $\alpha + \beta$

Fig. 3. ROC curves showing classification performance on SCOP classes.

4.2. *Prediction of protein function*

The performance of each property kernel when predicting whether a protein is annotated with the catalytic activity term is shown in the first column of Table 3. Here we found that disorder-based predictions (VSL2B disorder: $AUC = 0.742$; PDB disorder: $AUC = 0.718$) and predicted B-factors ($AUC = 0.722$) performed the best, whereas, contrary to their performance in distinguishing between SCOP categories, predicted secondary structures performed the worst (helix: $AUC = 0.687$; sheet: $AUC = 0.681$; loop: $AUC = 0.698$).

Table 3 also shows the aggregate performance of each property when predicting and testing on six subclasses of catalytic activity. As with predicting catalytic activity in general, predicted B-factors also performed well in predicting catalytic activity subclasses ($AUC^{\mathbf{w}} = 0.659$). Predicted secondary structures had mixed performance, with predicted loops obtaining a comparatively high $AUC^{\mathbf{w}}$ of 0.647, whereas helix and sheet predictions only achieved an $AUC^{\mathbf{w}}$ of 0.611 and 0.621, respectively.

We also generated a reduced redundancy data set of proteins with GO annotations in which the maximum pairwise sequence identity outputted by BLAST between any two sequences was

Table 3. Performance, according to AUC and $AUC^{\mathbf{w}}$, of each property-based feature when predicting catalytic activity and catalytic activity subclass, respectively. For each property feature the combination of m and n values that obtained the highest AUC are reported.

Property\Category	Catalytic activity			Catalytic subclass		
	m	n	AUC	m	n	$AUC^{\mathbf{w}}$
B-factors	256	32	0.722	–	–	0.659
Helix	256	8	0.687	–	–	0.611
Hydrophobicity	4,096	2	0.701	–	–	0.653
Loop	256	32	0.698	–	–	0.647
PDB disorder	256	32	0.718	–	–	0.650
Sheet	256	4	0.681	–	–	0.621
VSL2B disorder	256	16	0.742	–	–	0.620

Table 4. Performance, according to AUC and $AUC^{\mathbf{w}}$, of each the string kernel and combination of properties when predicting catalytic activity and catalytic activity subclass respectively. Results are shown for the full (redundant) data set and the non-redundant 40% data set (NR40).

Property\Category	Catalytic activity						Catalytic subclass					
	Full data set			NR40			Full data set			NR40		
	m	n	AUC	m	n	AUC	m	n	$AUC^{\mathbf{w}}$	m	n	$AUC^{\mathbf{w}}$
String kernel	-	5	0.857	-	5	0.733	-	5	0.930	-	5	0.649
VQ kernel	256	16	0.776	256	16	0.775	4,096	32	0.767	4.096	32	0.583
VQ + String kernel	-	-	0.781	-	-	0.775	-	-	0.767	-	-	0.585

at most 40%. This non-redundant data set (NR40) was generated to estimate the performance of each property when, for a given query protein, there is no sequence that is both annotated and of a reasonable level of sequence similarity. As shown by Figure 4 and Table 4, the performance of the property kernels was unaffected by the reduction in sequence identities between pairs of proteins, whereas string kernel performance was reduced.

4.3. *String kernel performance*

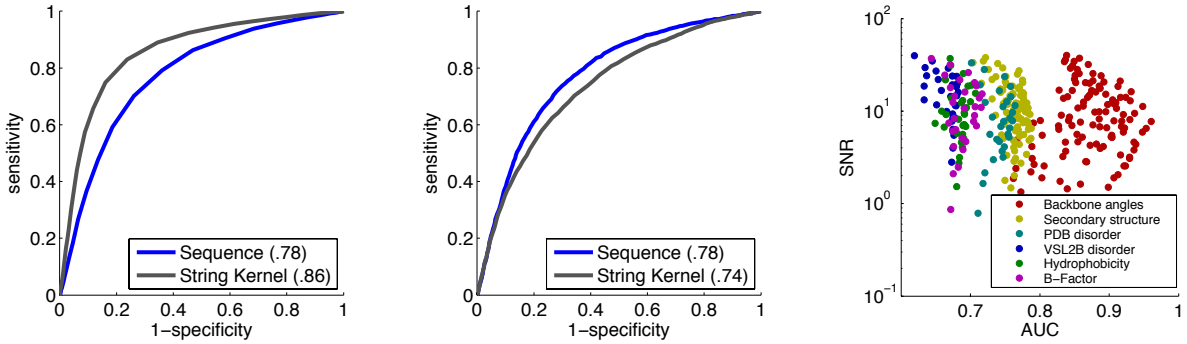
The string kernel did not show superior performance to any of the property kernels (both based on sequence and structure data) when predicting SCOP categories, only obtaining an $AUC^{\mathbf{w}}$ of 0.794 compared to an $AUC^{\mathbf{w}}$ of 0.961 obtained by the combined structure kernel and $AUC^{\mathbf{w}}$ of 0.813 obtained by the combined sequence kernel.

The performance of the string kernel in the task of function prediction was influenced by data set redundancy. When using redundant data, we found that the string kernel outperformed sequence-based properties in both the task of predicting catalytic activity and its subclass ($AUC^{\mathbf{w}}$ of 0.857 and 0.930), respectively (Figure 4(a)). However, when the redundancy in the protein function data was removed, the relative performance between the string kernel and the vector quantization kernel has reversed. As shown by Figure 4(b) the combined sequence-based property kernel achieved an AUC of 0.775 compared to 0.733 for the string kernel approach. Interestingly, this trend did not hold for the subclasses of catalytic activity, potentially due to the reduced data set sizes used to train individual models.

4.4. *Optimal parameter values*

We found that structure-based properties consistently preferred large numbers of centroids, obtaining maximum AUC at $m = 4096$ for all structure-based properties and all classification tasks. Optimal window sizes were 8 or 16 amino acids for most SCOP classes. Sequence-based properties were less consistent in the best-performing values of m and n, covering a range of values for each feature and SCOP class.

There was very little variation in preferred values of m when predicting catalytic activity with all features aside from predicted hydrophobicity obtaining maximum AUC values at $m = 256$. There was much more variation in preferred window sizes with hydrophobicity obtaining smallest optimal window size of 2, and B-factor, loop and PDB disorder predictions preferring longer window sizes ($n = 32$). Sequence based properties were much more consistent in the preferred values of m and n when predicting catalytic activity subclass, almost always favoring large values of m $(4,096)$.



(a) Catalytic activity (b) Catalytic activity (NR40) (c) SNR vs. $AUC^{\mathbf{w}}$

Fig. 4. Figure 4(a) shows ROC curves obtained when predicting catalytic activity using sequence properties (blue curve) and the string kernel (grey curve). AUC values are shown in parentheses in the figure legend. Figure 4(b) shows ROC curves obtained when predicting catalytic activity on the 40% non redundant data set of proteins (NR40) using sequence properties (blue) and the string kerenl (grey). Figure 4(c) shows obtained SNR values plotted as a function of $AUC^{\mathbf{w}}$ values for the prediction of SCOP class for each feature type.

4.5. *Comparing* AUC *and* SNR

Figure 4(c) shows a scatterplot of SNR and $AUC^{\mathbf{w}}$ values. Although, as a class, dihedral angles obtained higher values of $AUC^{\mathbf{w}}$, these values were only weakly correlated with higher SNR values ($\rho = 0.07$). For all other groups of properties in Figure 4(c) we observed a negative correlation between $AUC^{\mathbf{w}}$ and SNR.

5. Discussion

This paper introduced vector quantization (VQ) kernels and investigated their usefulness in different protein classification tasks. Several results show that the proposed kernel holds potential both as a standalone approach in protein classification and, more importantly, as a method that can be integrated into other strategies. The VQ kernel performed particularly

well in classification of SCOP classes, and as such could be readily exploited to automate the process of assigning new protein structures to structural classes. Such a method, similar to the FragBag approach,[34] is likely to be significantly faster than structure alignments that are commonly used for this purpose. Comparatively lower performance was observed in experiments relying on sequence-based properties only. Unsurprisingly, in these experiments, the property kernels outperformed string kernels when applied to non-redundant proteins, while they exhibited inferior performance to string kernels when high sequence identities were allowed.

The usefulness and biological significance of representing a protein sequence in a time series form has been long known. To the best of our knowledge, the use of a hydrophobicity plot (also referred to as hydropathy profile) was introduced by Rose who suggested that the local maxima and minima in the hydropathy profile typically correspond to the hydrophobic core and turns, respectively, in a protein's structure.[35] This idea quickly evolved into a tool for analysis of general properties of proteins, such as globular conformations[36] or membrane-spanning domains.[31] Advanced methods, such as the alignment of hydrophobic profiles[37] and Fast Fourier Transform (FFT) kernel[17] approach, have been proposed more recently, both in the context of recognizing membrane proteins.

The FFT kernel method is most related to the VQ kernels introduced here. In this method, Lanckriet and colleagues[17] first apply a low-pass filter to the original hydropathy profiles, pad the shorter profile with zeros (if the profiles are of different lengths), and subsequently calculate the kernel value between two FFT-derived spectra using a Gaussian kernel function with a free parameter σ. While this method provided solid performance in the task of predicting membrane proteins, we believe the kernel method introduced here offers better interpretability of results (through the selection and analysis of centroids) and more room for further refinements. For example, the simple inner product function between the count vectors $k(\mathbf{p}, \mathbf{q}) = \mathbf{x}^T\mathbf{y}$ can be augmented by a positive semi-definite matrix $\mathbf{Q}$ into a more general form $\mathbf{x}^T\mathbf{Q}\mathbf{y}$, perhaps by defining $\mathbf{Q}$ through a non-singular matrix of similarities between centroids ($\mathbf{S}$) and using $\mathbf{Q} = \mathbf{S}^T\mathbf{S}$. In addition, the centroid selection can be combined with motif discovery in time series data.[20] In terms of time complexity, the FFT kernel can be computed in $O(\ell \log \ell)$ time compared to $O(\ell \log m)$ time for the VQ kernel, where ℓ is the length of the protein and m the number of clusters. The VQ kernel may also hold promise to more easily integrate multiple types of properties and exploit their correlation via a joint clustering or some form of "matrix quantization".

In summary, the VQ kernel introduced in this work is a robust methodology that can easily be extended to any type of data that is or can be transformed into a time-series.

Acknowledgments

This work was funded by the National Science Foundation grant DBI-0644017.

References

1. P. Radivojac *et al.*, *Nat Methods* **10**, 221 (2013).
2. Y. Moreau and L.-C. Tranchevent, *Nat Rev Genet* **13**, 523 (2012).

3. B. Schölkopf, K. Tsuda and J.-P. Vert (eds.), *Kernel methods in computational biology* (The MIT Press, 2004).
4. W. S. Noble, Support vector machine applications in computational biology, in *Kernel methods in computational biology*, eds. B. Schölkopf, K. Tsuda and J.-P. Vert (The MIT Press, 2004) pp. 71–92.
5. C. Leslie *et al.*, *Pac Symp Biocomput* **575**, 564 (2002).
6. J. Qiu *et al.*, *Bioinformatics* **23**, 1090 (2007).
7. J.-P. Vert, *Bioinformatics* **18**, S276 (2002).
8. K. M. Borgwardt *et al.*, *Bioinformatics* **21**, i47 (2005).
9. A. Ben-Hur and W. S. Noble, *Bioinformatics* **21**, i38 (2005).
10. T. De Bie *et al.*, *Bioinformatics* **23**, i125 (2007).
11. L. Ralaivola *et al.*, *Neural Netw* **18**, 1093 (2005).
12. V. Vacic *et al.*, *J Comput Biol* **17**, 55 (2010).
13. T. Jaakkola *et al.*, Using the Fisher kernel method to detect remote protein homologies, in *Proc Int Conf Intell Syst Mol Biol, ISMB*, 1999.
14. T. Jaakkola *et al.*, *J Comput Biol* **7**, 95 (2000).
15. R. Kuang *et al.*, Profile-based string kernels for remote homology detection and motif extraction, in *Proc IEEE Computat Syst Bioinform Conf, CSB*, 2004.
16. S. V. N. Vishwanathan *et al.*, *J Mach Learn Res* **11**, 1201 (2010).
17. G. R. G. Lanckriet *et al.*, *Bioinformatics* **20**, 2626 (2004).
18. A. Sokolov and A. Ben-Hur, *J Bioinform Comput Biol* **8**, 357 (2010).
19. J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis* (Cambridge University Press, 2004).
20. P. Patel *et al.*, Mining motifs in massive time series databases, in *Proc IEEE Int Conf Data Mining, ICDM*, 2002.
21. Y. Linde *et al.*, *IEEE Trans Commun* **28**, 84 (1980).
22. A. Gersho and R. M. Gray, *Vector quantization and signal compression* (Kluwer Academic Publishers, 1992).
23. T. Tuytelaars and C. Schmid, Vector quantizing feature space with a regular lattice, in *Proc IEEE Int Conf Computer Vision, ICCV*, 2007.
24. G. Strang, *Introduction to linear algebra* (Wellsley-Cambrige Press, 2003).
25. M. Gönen and E. Alpaydin, *J Mach Learn Res* **12**, 2211 (2011).
26. A. G. Murzin *et al.*, *J Mol Biol* **247**, 536 (1995).
27. M. Ashburner *et al.*, *Nat Genet* **25**, 25 (2000).
28. A. Bairoch *et al.*, *Nucleic Acids Res* **33**, D154 (2005).
29. T. Joachims, *Learning to classify text using support vector machines: methods, theory, and algorithms* (Kluwer Academic Publishers, 2002).
30. W. Kabsch and C. Sander, *Biopolymers* **22**, 2577 (1983).
31. J. Kyte and R. F. Doolittle, *J Mol Biol* **157**, 105 (1982).
32. P. Radivojac *et al.*, *Protein Sci* **13**, 71 (2004).
33. K. Peng *et al.*, *BMC Bioinformatics* **7**, 208 (2006).
34. I. Budowski-Tal *et al.*, *Proc Natl Acad Sci U S A* **107**, 3481 (2010).
35. G. D. Rose, *Nature* **272**, 586 (1978).
36. G. D. Rose and S. Roy, *Proc Natl Acad Sci U S A* **77**, 4643 (1980).
37. J. S. Lolkema and D.-J. Slotboom, *FEMS Microbiol Rev* **22**, 305 (1998).