2D similarity kernels for biological sequence classification

Pavel P. Kuksa
Machine Learning Department
NEC Laboratories America, Inc
Princeton, NJ, USA
pkuksa@nec-labs.com

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 tasks such as document topic elucidation, biological sequence classification, or protein superfamily and fold prediction. However, typical string kernel methods rely on analysis of discrete 1D string data (e.g., DNA or amino acid sequences). This work introduces new 2D kernel methods for sequence data in the form of sequences of feature vectors (as in biological sequence profiles, or sequences of individual amino acid physico-chemical descriptors). On three protein sequence classification tasks proposed 2D kernels show significant 15-20% improvements compared to state-of-the-art sequence classification methods.

Keywords

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 document and text classification 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, 6, 12, 9, 17]. These algorithms often exhibit state-of-the-art performance on tasks such as protein superfamily and fold prediction [6, 9, 10, 14], or DNA sequence analysis [8].

A family of state-of-the-art approaches to scoring similarity between pairs of sequences relies on fixed length, substring spectral representations and the notion of mismatch kernels, c.f. [6, 12]. There, a sequence is represented as the spectra (counts) of all short substrings (k-mers) contained within a sequence. The similarity score is established by exact or approximate matches of k-mers. Initial work, e.g., [12, 16], 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, [7] introduced linear time algorithms with alphabet-independent complexity applicable to computation of a large class of existing string kernels.

However, typical spectral models (e.g., mismatch/spectrum kernels, gapped and wildcard kernels [10, 12]) essentially rely on *symbolic Hamming-distance* based matching of 1D k-mers. For example, given a 1D sequence X over alphabet Σ the *spectrum-k* kernel [11] and the *mismatch-*(k,m) kernel [12] induce $|\Sigma|^k$-dimensional representation

$$\Phi_{k,m}(X) = \left(\sum_{\alpha \in X} I_m(\alpha, \gamma) \right)_{\gamma \in \Sigma^k} \quad (1)$$

where $I_m(\alpha,\gamma) = 1$ if $\alpha \in N_{k,m}(\gamma)$, and $N_{k,m}(\gamma)$ is the *mutational neighborhood* of γ, the set of all k-mers that differ from γ by at most m mismatches.

We note that existing k-mer string kernels essentially use only *1D discrete* sequences and Hamming-based matching, while input data may often be represented in the form of sequences of *R-dim. (real-valued) feature vectors* (2D sequences). This is the case, for instance, in commonly used profile representations [6, 4] of amino acid sequence in biological sequence analysis, or representations of proteins as sequences of individual amino acid physico-chemical descriptors [18, 20]; such feature sequences (2D) can provide richer and more accurate representations for biological sequences [6, 18].

In this work, we consider an approach that directly exploits these richer R-dim. feature sequences (e.g., sequence profiles or descriptor sequences) and propose general, simple *2D representations* of sequences (Sec. 3). We then introduce a class of *2D similarity kernels* that allows efficient inexact matching and classification of sequence inputs in the form of *sequences of R-dim. feature vectors* (Sec. 3.3). The developed approach is applicable to modeling of both *discrete-* and *continuous*-valued sequences, such as biological sequence profiles, or sequences of amino acid descriptors. Experiments using the new 2D kernels on protein remote homology detection and fold prediction show excellent predictive performance (Sec. 4) with significant *15%-20%* improvements in predictive accuracy over the 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 [19] methods provide some of the most accurate results [6, 12, 17, 10, 9, 2].

The key idea of basic string kernel methods is to apply a mapping $\Phi(\cdot)$ to map sequences of variable length into a fixed-dimensional vector space. In this space a standard classifier such as a support vector machine (SVM) [19] 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:

$$K(X, Y) = \langle \Phi(X), \Phi(Y) \rangle, \quad (2)$$

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 co-occurrence of exact sub-patterns (k-mers, features), as in spectrum kernels [11] or substring kernels [21]. Inexact comparison in this framework is typically achieved using different families of mismatch [12] or profile [6] 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 [6], builds a profile [4] 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. 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. 2D sequence). For example, protein sequences could be considered as 2D sequences of R-dim. feature vectors describing physical/chemical properties of individual amino acids, or as *sequence profiles* describing each sequence position with probability distribution over amino acid characters.

In this work, we aim at methods that directly exploit these richer 2D sequence representations (e.g., profiles or amino acid descriptors) to improve accuracy and propose a family of 2D similarity kernels(Sec. 3, 3.3) that as we show empirically (Sec. 4) provide effective improvements in practice over traditional 1D sequence kernels for a number of challenging sequence classification problems.

3. 2D SEQUENCE REPRESENTATIONS

In a typical setting, string kernels are applied to 1D string data, e.g., amino acid sequences or DNA sequences. In this work we consider alternative *2D representations* for sequences (Fig. 1a) as *sequences of R-dim. feature vectors*. In particular, we consider two representations:

1) *Symbolic embedding.* Encoding original continuous-valued R-dim. feature vectors in discrete (binary) E-dim. space using e.g. similarity hashing approach [22] (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) $R \times |X|$ feature sequence X is re-represented as $E \times |X|$ or $|R| \times |X|$ 2D discrete feature sequence.

We will show in the experiments that using these *2D representations* can *significantly (by 15-20%) improve* predictive accuracy compared to traditional 1D 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 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 of the feature data range could also be obtained by using 1D clustering, e.g. k-means, to adaptively choose dicretization levels.

3.2 Discrete (symbolic) Embedding

Given input sequence $X = x_1, \dots, x_n$ of R-dim. feature vectors, each R-dim. vector could be mapped into discrete feature vectors using symbolic embedding $E(\cdot)$ as in, for example, similarity hashing [22]. Using similarity hashing, input sequence $X = x_1, ..., x_n$ of R-dimensional feature vectors is mapped intro a *binary* Hamming-space embedded sequence

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

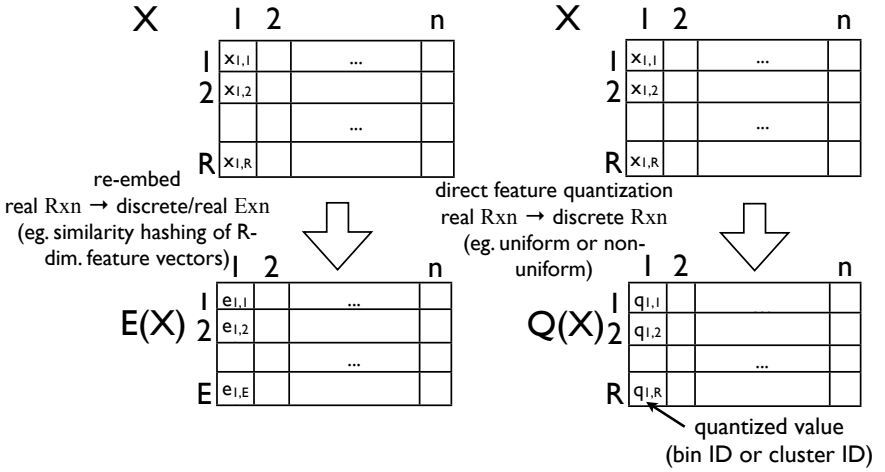
where $E(x_i) = e^i_1 e^i_2 \dots e^i_B$ is a symbolic Hamming embedding for item x_i in X, with $|E(x_i)| = B$, the number of bits in a resulting binary embedding of x_i. This embedding as proposed in [22] essentially aims to minimize average Hamming distance between binary embeddings corresponding to similar R-dim. data points:

$$\min \sum_{\alpha,\beta} S(\alpha, \beta) d(E(\alpha), E(\beta))$$

Under this embedding, the Hamming similarity, $h_{\alpha,\beta}$, 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. feature vectors α and β:

$$h(E(\alpha), E(\beta)) \propto \mathcal{S}(\alpha, \beta). \quad (3)$$

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



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

Figure 1: Proposed *2D representations*.

3.3 2D similarity kernels

We now introduce efficient 2D kernels for the proposed 2D sequence representations.

Similarity evaluation between two 2D sequences X and Y under 2D *matrix representations* amounts to comparing pairs of 2D submatrices contained in X and Y. A 2D string kernel can be defined for 2D sequences X and Y as

$$K_{2D}(X,Y) = \sum_{\alpha_{2D} \in X} \sum_{\beta_{2D} \in Y} \mathcal{K}(\alpha_{2D}, \beta_{2D}) \qquad (4)$$

where α_{2D} and β_{2D} are $|R| \times k$ (or $|E| \times k$) submatrices of X and Y and $\mathcal{K}(\alpha_{2D}, \beta_{2D})$ is a kernel function defined for measuring similarity between two submatrices. One possible definition for $\mathcal{K}(\cdot,\cdot)$ that we use in this work is row-based similarity

$$\mathcal{K}(\alpha_{2D}, \beta_{2D}) = \sum_{i=1}^{R} I(\alpha_{2D}^i, \beta_{2D}^i) \qquad (5)$$

where $I(\cdot,\cdot)$ is a similarity/indicator function for matching 1D rows α_{2D}^i and β_{2D}^i. The matching function $I(\cdot,\cdot)$ could be defined as $I(\alpha,\beta) = 1$ if $d(\alpha,\beta) \leq m$, and 0 otherwise (similar to the mismatch kernel).

Using Eq. 5, 2D kernel in Eq. 4 can be written as

$$K_{2D}(X,Y) = \sum_{i=1}^{R} \sum_{\alpha_{2D} \in X} \sum_{\beta_{2D} \in Y} I(\alpha_{2D}^i, \beta_{2D}^i) \qquad (6)$$

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 2D kernel 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 sequence classification problems using standard benchmark datasets for protein sequence analysis.

Datasets and experimental setup. We test proposed methods on a number of multi-class sequence classification tasks: (1) protein remote homology detection (SCOP dataset with 7329 sequences) [23, 6] (2) multi-class protein fold recognition (Ding-Dubchak dataset, 27 protein folds, 694 sequences) [3, 14], and (3) multi-class remote fold recognition [14].

We also compare with a number of other state-of-the-art methods for sequence classification. For remote homology and protein fold prediction tasks, we use BLOSUM amino acid substitution vectors to obtain a 2D amino acid sequence representation. We also use 2D sequence profiles and compare with profile kernel approach [6].

We use state-of-the-art spectrum/mismatch [10] and spatial (SSSK) [9] kernels as our basic 1D similarity kernels. 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. 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/2Dstring.html`.

Evaluation measures. For 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 misclassifying 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). 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.

4.1 Remote homology detection

For the task of remote homology detection (Table 1), we compare our proposed 2D string kernel method (using BLOSUM rows as feature vectors for individual amino acids, i.e. $20 \times |X|$ 2D sequence) with a number of state-of-the-art kernel methods for remote homology including spectrum/mismatch kernels [12, 11], spatial sample kernels [9], semi-supervised cluster kernel [23], as well as state-of-the-art profile kernel [6]. We also compare with a recently proposed spectrum-RBF and mismatch-RBR methods [18] which incorporate physico-chemical descriptors with traditional spectrum/mismatch kernels. We test both similarity hashing and direct feature quantization approaches with our 2D string kernels.

As can be seen from results in Table 1, 2D 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%). 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 2D kernel provides substantial improvements in semi-supervised settings using semi-supervised cluster kernel [23] and profile kernel approaches. For example, 2D kernel on sequence profiles used by the profile kernel (obtained from non-redundant sequence database (NRDB) [6]) achieves higher average ROC50 score of 86.27 compared to 81.51 of the profile kernel. We also note the our 2D string kernel using only BLOSUM substitution scores achieves higher average ROC50 scores compared to computationally more expensive spectrum-RBF/mismatch-RBF approaches [18] which exploit richer descriptors (BLOSUM, AAindex descriptors, etc). We also note that using direct feature quantization provides more effective improvements compared to discrete embedding with similarity hashing.

4.2 Multi-class protein fold prediction

For multi-class protein fold recognition (Table 2, Ding&Dubchack dataset, 27-folds), using 2D 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 [10] which also uses BLOSUM substitution scores to measure similarity between k-mers.

On a challenging remote fold recognition dataset [14] (results in Table 3), we observe similar improvements in ranking quality when using 2D string 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.

4.3 Running time

In Table 4, we compare the running time for the proposed 2D string kernel and traditional string kernel methods. We note that for mismatch-(k,m) kernel computation (protein remote homology data) we use linear time sufficient-statistic based algorithm from [7]. As can be seen from results, using 2D kernels gives similar performance in running times compared to traditional 1D kernels while displaying better classification performance (Table 1).

5. CONCLUSIONS

We presented new 2D kernel methods for biological sequences represented as 'sequences of feature vectors (as in biological sequence profiles, or sequences of amino acid descriptors). The proposed approach directly exploits these feature sequences (2D) to improve sequence classification. On three protein sequence classification tasks this shows significant 15-20% improvements compared to state-of-the-art sequence classification methods.

References

[1] Jianlin Cheng and Pierre Baldi. A machine learning information retrieval approach to protein fold recognition. *Bioinformatics*, 22(12):1456–1463, June 2006.

[2] Corinna Cortes, Patrick Haffner, and Mehryar Mohri. Rational kernels: Theory and algorithms. *Journal of Machine Learning Research*, 5:1035–1062, 2004.

[3] Chris H.Q. Ding and Inna Dubchak. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17(4):349–358, 2001.

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

[5] Tony Jebara, Risi Kondor, and Andrew Howard. Probability product kernels. *J. Mach. Learn. Res.*, 5:819–844, December 2004.

[6] Rui Kuang, Eugene Ie, Ke Wang, Kai Wang, Mahira Siddiqi, Yoav Freund, and Christina S. Leslie. Profile-based string kernels for remote homology detection and motif extraction. In *CSB*, pages 152–160, 2004.

[7] Pavel Kuksa, Pai-Hsi Huang, and Vladimir Pavlovic. Scalable algorithms for string kernels with inexact matching. In *NIPS*, 2008.

[8] Pavel Kuksa and Vladimir Pavlovic. Efficient alignment-free dna barcode analytics. *BMC Bioinformatics*, 10(Suppl 14):S9, 2009. Impact factor: 3.78.

[9] Pavel P. Kuksa and Vladimir Pavlovic. Spatial representation for efficient sequence classification. In *ICPR*, 2010.

Table 1: Classification performance (mean ROC50) on protein remote homology detection (54 experiments)

Method	Mean ROC50
Baseline 1: Spectrum [11]	27.91
Baseline 2: Mismatch-(k=5,m=1) [12]	41.92
Baseline 3: Spatial sample (SSSK) [9]	50.12
Baseline 4: Spectrum-RBF [18]	42.1
Baseline 5: Mismatch-RBF [18]	43.6
Baseline 6: Semi-supervised Cluster kernel [23]	67.91
Baseline 7: Profile kernel-(k=5,σ=7.5) (NRDB) [6]	81.51
Spectrum (2D BLOSUM, sim. hashing)	38.68
Mismatch (2D BLOSUM, sim. hashing)	44.05
Spectrum (2D BLOSUM)	43.29
Mismatch (2D BLOSUM)	49.17
Spatial sample (SSSK) (2D BLOSUM)	55.54
Semi-supervised Cluster kernel (2D BLOSUM)	70.14
Profile kernel (2D)	86.27

Table 2: Multi-class protein fold prediction [3] (27-class)

Method	Error, %	Balanced error, %	F1
Baseline 1: Mismatch-(k=5,m=1)	51.17	53.22	61.68
Baseline 2: Substitution kernel [10]	45.43	48.02	53.54
Spectrum (2D BLOSUM)	43.86	48.49	63.18

Table 3: Classification performance on fold prediction (multi-class) [14]

Method	Error	Top 5 Error	Balanced Error	Top 5 Balanced Error	F1	Top 5 F1
Baseline 1: PSI-BLAST [14]	64.80	51.80	70.30	54.30	-	-
Baseline 2: Substitution kernel [10] (BLOSUM62) [10]	51.95	27.04	66.17	36.72	34.49	66.27
Baseline 3: Spatial sample kernel (SSSK) [9]	48.7	25.08	73.04	44.05	30.57	62.37
Baseline 4: Mismatch-(k=5,m=1) [12]	53.75	29.15	82.75	52.40	16.92	56.67
Baseline 5: Profile (5,7.5) (Swiss-prot) [6]	49.35	20.36	76.67	35.28	26.05	68.09
Baseline 6: Semi-supervised Cluster kernel (Swiss-Prot) [23]	48.86	19.54	72.88	34.06	26.59	70.07
Mismatch (k=5,m=1) (2D BLOSUM)	52.12	24.10	81.76	43.30	22.74	64.30
Spatial sample kernel (2D BLOSUM+Hydropathy)	47.88	19.38	70.81	30.99	32.86	74.57
Semi-supervised Cluster kernel (Swiss-Prot) (2D BLOSUM)	48.86	18.40	74.87	28.92	27.06	74.24

Table 4: Running time for the kernel computations

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

[10] Christina Leslie and Rui Kuang. Fast string kernels using inexact matching for protein sequences. *J. Mach. Learn. Res.*, 5:1435–1455, 2004.

[11] Christina S. Leslie, Eleazar Eskin, and William Stafford Noble. The spectrum kernel: A string kernel for SVM protein classification. In *Pacific Symposium on Biocomputing*, pages 566–575, 2002.

[12] Christina S. Leslie, Eleazar Eskin, Jason Weston, and William Stafford Noble. Mismatch string kernels for SVM protein classification. In *NIPS*, pages 1417–1424, 2002.

[13] Zhiwu Lu and H.H.S. Ip. Image categorization with spatial mismatch kernels. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:397–404, 2009.

[14] Iain Melvin, Eugene Ie, Jason Weston, William Stafford Noble, and Christina Leslie. Multi-class protein classification using adaptive codes. *J. Mach. Learn. Res.*, 8:1557–1581, 2007.

[15] Protein fold prediction data set. http://ranger.uta.edu/ËIJchqding/bioinfo.html.

[16] John Shawe-Taylor and Nello Cristianini. *Kernel Meth-*

ods for Pattern Analysis. Cambridge University Press, New York, NY, USA, 2004.

[17] Sören Sonnenburg, Gunnar Rätsch, and Bernhard Schölkopf. Large scale genomic sequence SVM classifiers. In *ICML '05*, pages 848–855, New York, NY, USA, 2005.

[18] Nora Toussaint, Christian Widmer, Oliver Kohlbacher, and Gunnar Ratsch. Exploiting physico-chemical properties in string kernels. *BMC Bioinformatics*, 11(Suppl 8):S7, 2010.

[19] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998.

[20] Mathura S. Venkatarajan and Werner Braun. New quantitative descriptors of amino acids based on multidimensional scaling of a large number of physicalâĂŞchemical properties. *Molecular modeling annual*, 7:445–453, 2001.

[21] S. V. N. Vishwanathan and Alex Smola. Fast kernels for string and tree matching. In *NIPS*, 2002.

[22] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1753–1760. 2009.

[23] Jason Weston, Christina Leslie, Eugene Ie, Dengyong Zhou, Andre Elisseeff, and William Stafford Noble. Semi-supervised protein classification using cluster kernels. *Bioinformatics*, 21(15):3241–3247, 2005.

[24] Dell Zhang, Xi Chen, and Wee Sun Lee. Text classification with kernels on the multinomial manifold. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 266–273, New York, NY, USA, 2005. ACM.