Fast Protein Homology and Fold Detection with Sparse Spatial Sample Kernels

Pavel Kuksa, Pai-Hsi Huang, Vladimir Pavlovic
Department of Computer Science
Rutgers University, Piscataway NJ 08854
Email: {pkuksa;paihuang;vladimir}@cs.rutgers.edu

Abstract

In this work we present a new string similarity feature, the sparse spatial sample (SSS). An SSS is a set of short substrings at specific spatial displacements contained in the original string. Using this feature we induce the SSS kernel (SSSK) which measures the agreement in the SSS content between pairs of strings. The SSSK yields better prediction performance at substantially reduced computational cost than existing algorithms for sequence classification tasks. We show that on the task of predicting the functional and structural classes of proteins, the SSSK results in state-of-the-art performance across several benchmark sets in both supervised and semi-supervised learning settings. The results have immediate practical value for accurate protein superfamily and fold classification and may be similarly extended to other sequence modeling domains.

1 Introduction

Structural or functional classification of proteins is a fundamental problem in computational biology. With more than 61 million sequences in GenBank [3] and 5.3 million unannotated sequences in UNIPROT [2], experimental elucidation of an unknown protein function becomes expensive, making development of computational aids for sequence annotation based on primary sequences only a critical and timely task. In this work, we focus on the problem of predicting protein remote homology (superfamily) and fold using the primary sequence information.

Developments in computationally-aided protein functional or structural annotation in the past decade have witnessed the benefit of discriminative modeling methods that outperform traditional generative sequence models. In the generative model setting, the goal is to capture the commonly shared characteristics within the group, or class, using only positive training examples. Examples of methods that operate under this setting are PSI-BLAST [1], profiles [6], and profile hidden Markov models (profile HMMs) [5]. However, the shared characteristics may also be present in other groups of interest, and therefore may lead to sub-optimal classification accuracy. The discriminative models, on the other hand, focus on capturing the differences between groups using both positive and negative examples. The discriminative method such as kernel-based [16] machine learning methods provide some of the most accurate results [9, 11, 15] in many sequence analysis tasks. Jaakkola *et al*. proposed the *SVMFisher* in [8] with features derived from a probabilistic model. Leslie *et al*. in [11] proposed a class of kernel methods that operate directly on strings and derive features from the sequence content. Both classes of kernels demonstrated improved discriminative power over generative methods.

In this study we propose the *sparse spatial sample* (SSS) features and induce a new family of *sparse spatial sample kernels* (SSSK) for sequence analysis tasks. The proposed kernels *explicitly* model biological transformations such as mutation, insertion and deletion while incurring low computation cost, compared to other state-of-the-art methods. In contrast to the existing string kernels, the SSSK provide a richer representation for sequences by explicitly encoding the information on spatial configuration of features within the sequence. The proposed methods perform significantly better and run substantially faster than existing state-of-the-art algorithms, including the profile [9] and neighborhood mismatch [17] kernels.

2 The Sparse Spatial Sample Features and Kernels

We define the family of SSS features as the substrings of type $a_1 \overset{d_1}{\leftrightarrow} a_2, \overset{d_2}{\leftrightarrow}, \cdots, \overset{d_{t-1}}{\longleftrightarrow} a_t$ (a_1 separated by d_1 characters from a_2, a_2 separated by d_2 characters from a_3, etc.) contained in sequence X. This is illustrated in Figure 1. The kernel SSSK is then induced by

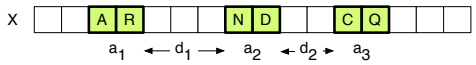


Figure 1. The SSS feature.

matching the cumulative (spectral) SSS content of two sequences X and Y. Parametrized by three positive integers, the proposed family of kernels has the following form:

$$K^{(t,k,d)}(X,Y) = \sum_{\substack{(a_1,d_1,\ldots,d_{t-1},a_t) \\ a_i \in \Sigma^k, 0 \le d_i < d}} C(a_1,d_1,\cdots,a_{t-1},d_{t-1},a_t|X) \cdot C(a_1,d_1,\cdots,a_{t-1},d_{t-1},a_t|Y) \quad ,(1)$$

where $C(a_1,d_1,\cdots,a_{t-1},d_{t-1},a_t|X)$ denotes the number of times we observe the particular SSS feature in sequence X.

The new kernel implements the idea of sampling the sequences at different resolutions and comparing the resulting spectra; similar sequences will have similar spectrum at one or more resolutions. This takes into account possible mutations, as well as insertions/deletions. Each sample consists of t spatially-constrained probes of size k, each of which lie no more than d positions away from its neighboring probes. In the proposed kernels, the parameter k controls the individual probe size, d controls the locality of the sample, and t controls the cardinality of the sampling neighborhood. In this work, we use short samples of size 1 (i.e., $k = 1$), and set t to 2 (i.e. features are pairs of monomers) or 3 (i.e. features are triples.)

The proposed sample string kernels, not only take into account feature counts (as in the family of spectrum [11] or gapped/subsequence [10, 14] kernels), but also explicitly encode spatial configuration information, *i.e.* how the features are positioned in the sequence. The spatial information can be critical in establishing similarity of sequences under complex transformations such as the evolutionary processes in protein sequences. The addition of the spatial information experimentally demonstrates very good performance, even with very short sequence features (*i.e. k=1*), as we will show in Section 3.

3 Experimental Results

We present experimental results for the protein remote homology (superfamily) prediction under the supervised and semi-supervised settings on the SCOP [13] dataset in Section 3.1 and the results for remote fold recognition in Section 3.2. In all experiments, all kernel values $K(X,Y)$ are normalized using $K'(X,Y) = \frac{K(X,Y)}{\sqrt{K(X,X)K(Y,Y)}}$ to remove the dependency between the kernel value and the sequence length.

3.1 Remote homology detection

We perform our experiments for remote homology detection on a widely used benchmark SCOP 1.59 dataset [17, 9], which contains $7,329$ protein sequences and 54 binary classification problems, each simulating protein remote homology detection by completely holding out one family in a superfamily for testing. Only $2,862$ domains out of $7,329$ are labeled, which allows to perform experiments in both supervised and semi-supervised (labeled and unlabeled sequences) settings.

We evaluate all methods using the *Receiver Operating Characteristic* (ROC) and ROC-50 [7] scores. The ROC-50 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 ROC-50 score is typically more indicative of the prediction accuracy of a homology detection method than the ROC score.

In the semi-supervised experiments, we use kernel smoothing as in [17]. For each sequence X, we query the unlabeled dataset with PSI-BLAST and recruit the sequences with e-values ≤ 0.05 as the neighbors of X.

Supervised setting: We compare the performance of our proposed methods with previously published state-of-the-art methods [12, 11] under the supervised learning setting in Table 1. We also show the dimensionality of the induced features and the observed experimental running times, measured on a 2.8GHz CPU, for constructing the 7329x7329 kernel matrix[1]. It is clear from the table that the proposed kernels not only show significantly better performance than existing methods, but also require substantially less computational time. Also, as can be seen from the comparison with the gapped kernels, the addition of the spatial information substantially improves the classification performance.

We also show the ROC-50 plot in Figure 2(a). In the plot, the horizontal axis corresponds to the ROC-50

[1]The code used for evaluation of the competing methods has been highly optimized to perform on par or better than the published spectrum/mismatch code. We also used the code provided by the authors of the competing methods.

scores and the vertical axis denotes the number of experiments, out of 54, with an equivalent or higher ROC-50 score. Our results clearly indicate that both double and triple kernels dominate the mismatch(5,1) kernel, as well as other supervised methods.

Table 1. Comparison of the performance under the supervised setting.

Method	ROC	ROC50	# dim.	Time (s)
(5, 1)-mismatch	0.8749	0.4167	3200000	938
SVM-pairwise [12]	0.8930	0.4340	-	-
gapped(6,2) [10]	0.8296	0.3316	400	55
gapped(7,3)	0.8540	0.3953	8000	297
subsequence-2 [14]	0.8581	0.3583	400	133
subsequence-3	0.8723	0.4037	8000	1543
(1,5) double	0.8901	0.4629	2000	54
(1,3) triple	**0.9148**	**0.5118**	72000	112

Parameters of kernels in parenthesis indicate: the probe size (k) and the maximum distance between adjacent samples for the double and triple kernels; the length of the contiguous k-mer and the maximum number of mismatches for the mismatch kernel; the maximum window size and the length of the k-mer for the gapped kernels.

Table 2. Comparison of the performance under the semi-supervised setting.

Method	ROC	ROC50
(5, 1)-mismatch neighborhood	0.9093	0.6745
(5,7.5)-profile	0.9190	0.6069
(1,5)-double neighborhood	0.9282	0.6383
(1,3)-triple neighborhood	**0.9382**	**0.7262**

Table 3. Comparison on Ding and Dubchak benchmark data set

Method	Error	Top 5 Error	Balanced Error	Top 5 Balanced Error
SVM(D&D) [4]	-	-	56.5	-
Mismatch(5,1)	51.17	22.72	53.22	28.86
Double(1,5)	44.13	23.50	46.19	23.92
Triple (1,3)	**41.51**	**18.54**	**44.99**	**21.09**

Semi-supervised setting: We compare the performance on the same data set in Table 2 and in Figure 2(b) under the semi-supervised setting. The triple neighborhood kernel outperforms both the profile kernel and the mismatch neighborhood kernel, the state-of-the-art classifiers reported in previous studies [9, 17].

3.2 Remote fold recognition

For the fold recognition task, we use a challenging dataset designed by Ding *et al.* [2] in [4], a benchmark used in many studies. The data set contains sequences from 27 folds divided into two *independent* sets, with the training and test sequences sharing less than 35% sequence identities and within the training set, no sequences share more than 40% sequence identities. We compare the performance of our methods under supervised setting with previously published methods on Ding and Dubchak benchmark data set in Table 3. Our spatial kernels achieve higher performance compared to the state-of-the-art classifiers.

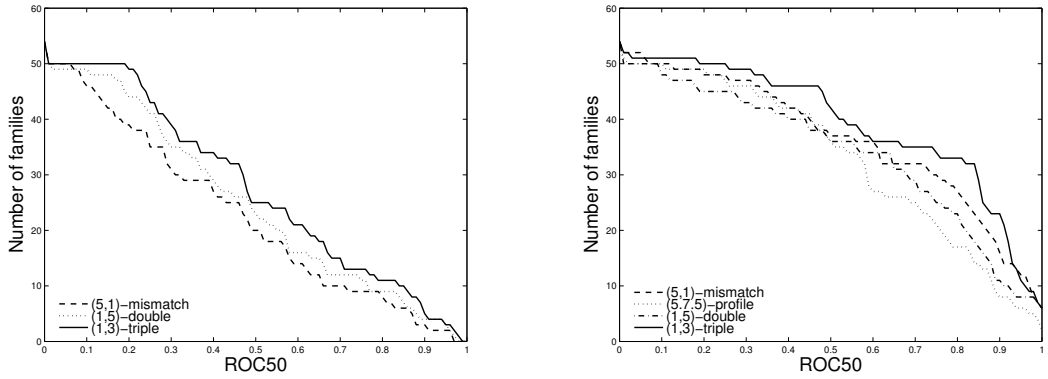
4 Discussion

Our family of kernels posses clear computational advantages over most existing methods. We next outline these advantages and point to a possible biological significance of the proposed SSS features.

Complexity Comparison: Both mismatch and profile kernels have higher complexity compared to the sample kernels. The total complexity for a set of N sequences of length n is $O(dnN + \min(u, dn)N^2)$ for doubles and $O(d^2nN + \min(u, d^2n)N^2)$ for triples, where u is the number of unique features. This can be significantly lower than the exponential complexity of the mismatch kernel $O(k^{m+1}|\Sigma|^m nN + min(u', n)N^2)$, where $u' \leq |\Sigma|^k$, $k = 5, 6$, and Σ is the alphabet set. This complexity difference leads to order-of-magnitude improvements in the running times of the sample kernels over the mismatch and profile kernels.

Biological Motivation: Compared to mismatch/profile kernels, the feature sets induced by our kernels cover segments of variable length (for example, 2-6 residues for the double-$(1, 5)$ kernel), whereas the mismatch and profile kernels cover contiguous fixed-length segments (e.g., 5 or 6 residues). The proposed features also capture short-term dependencies and interactions between local sequence features by explicitly encoding spatial information; in contrast, such information is not present in the gapped/subsequence kernels. As shown by our experiments in Table 1, the spatial information is crucial for accurate sequence classification (e.g. the gapped/subsequence kernels based on 2- and 3-mers show substantially lower classification performance).

[2] http://ranger.uta.edu/∼chqding/bioinfo.html



(a) SCOP dataset. Supervised setting.

(b) SCOP dataset. Semi-supervised setting

Figure 2. Comparison of the performance (ROC50) in a supervised setting (left) and in a semi-supervised setting (right) using SCOP 1.59 as the unlabeled dataset.

5 Conclusion

We present a new family of sparse spatial string features and kernels demonstrating state-of-the-art performance on protein remote homology and fold prediction, two important tasks in computational biology. The proposed methods have low computational cost and yield significantly improved homology and fold detection performance compared to other state-of-the-art approaches. The key component of the method is the spatially-constrained sample kernel for efficient sequence comparison leading to fast and accurate remote homology and fold detection. The proposed methodology can be applied to other challenging problems in sequence analysis, such as the text modeling, music classification, etc.

References

[1] S. Altschul et al. Gapped Blast and PSI-Blast: A new generation of protein database search programs. *NAR*, 25:3389–3402, 1997.

[2] A. Bairoch, R. Apweiler, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, D. A. Natale, C. O'Donovan, N. Redaschi, and L.-S. L. Yeh. The Universal Protein Resource (UniProt). *Nucl. Acids Res.*, 33(suppl-1):D154–159, 2005.

[3] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler. Genbank. *Nucl. Acids Res.*, 33(suppl-1):D34–38, 2005.

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

[5] S. Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9):755–763, 1998.

[6] M. Gribskov, A. McLachlan, and D. Eisenberg. Profile analysis: detection of distantly related proteins. *PNAS*, 84:4355–4358, 1987.

[7] M. Gribskov and N. L. Robinson. Use of receiver operating characteristic (roc) analysis to evaluate sequence matching. *Computers & Chemistry*, 20(1):25–33, 1996.

[8] T. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. In *J. Comp. Biol.*, volume 7, pages 95–114, 2000.

[9] R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. Leslie. Profile-based string kernels for remote homology detection and motif extraction. In *CSB*, pages 152–160, August 2004.

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

[11] C. S. Leslie, E. Eskin, J. Weston, and W. S. Noble. Mismatch string kernels for svm protein classification. In *NIPS*, pages 1417–1424, 2002.

[12] L. Liao and W. S. Noble. Combining pairwise sequence similarity and support vector machines for remote protein homology detection. In *RECOMB*, pages 225–232, 2002.

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

[14] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *J. Mach. Learn. Res.*, 2:419–444, 2002.

[15] S. Sonnenburg, G. Rätsch, and B. Schölkopf. Large scale genomic sequence svm classifiers. In *ICML*, pages 848–855, New York, NY, USA, 2005.

[16] V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998.

[17] J. Weston, C. Leslie, E. Ie, D. Zhou, A. Elisseeff, and W. S. Noble. Semi-supervised protein classification using cluster kernels. *Bioinformatics*, 21(15):3241–3247, 2005.