On the role of local matching for efficient semi-supervised protein sequence classification

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

Abstract

Recent studies in protein sequence analysis have leveraged the power of unlabeled data. For example, the profile and mismatch neighborhood kernels have shown significant improvements over classifiers estimated under the fully supervised setting. In this study, we present a principled and biologically motivated framework that more effectively exploits the unlabeled data by only utilizing regions that are more likely to be biologically relevant for better prediction accuracy. As overly-represented sequences in large uncurated databases may bias kernel estimations that rely on unlabeled data, we also propose a method to remove this bias and improve performance of resulting classifiers. Combined with a computationally efficient sparse family of string kernels, our proposed framework achieves state-of-the-art accuracy in semi-supervised protein remote homology detection on three large unlabeled databases.

1 Introduction

In this work we address the problem of predicting protein remote homology using only the primary sequence. This is a common and critical task that arises when other sources of information such as the secondary or tertiary structure are not available. Remote homology detection settings, such as the successful kernel-based methods [5, 8], are typically characterized by few *positive training* sequences accompanied by a large number of negative training examples. The lack of positive training examples may lead to sub-optimal classifier performance, prompting the need for expansion of the training set. However, enlarging the set by experimentally labeling the sequences is costly. Instead, one typically leverages *unlabeled data* to refine the decision boundary. State-of-the-art methods such as the profile [6] and the mismatch neighborhood kernel [11] both show significant gains in performance resulting from the use of these large but unlabeled data sources. However, the need for additional reduction in computational complexity and improvement in predictive accuracy hinders the widespread use of these powerful computational tools.

In this study, we propose a systematic and biologically motivated approach that more efficiently uses the unlabeled data and further develops the crucial aspects of neighborhood and profile kernels. The proposed framework, the *region-based neighborhood method* (Sec. 3.1), utilizes the unlabeled sequences to construct a more accurate classifier by focusing on the significantly similar sequence regions that are more likely to be biologically relevant as opposed to using whole sequences directly. As overly-represented sequences may lead to performance degradation by biasing kernel estimations based on unlabeled data, we propose an effective clustering method in Sec. 3.2 that improves performance of the resulting classifiers under the semi-supervised learning setting. Our experimental results show that the framework we propose yields significantly better performance compared to the state-of-the methods and also demonstrates significantly improved running times on large unlabeled datasets.

2 Background

The spectrum kernel family has become one of the most accurate tools for protein homology detection. Spectral methods rely on fixed-length representations or features $\Phi(X)$ of arbitrary long sequences X modeled as the spectra of short substrings (k-mers) contained in the sequences. These features are subsequently used to define the measure of similarity, or the kernel, $K(X,Y) = \Phi(X)'\Phi(Y)$ between pairs of sequences (X,Y). Remote homology settings rely on inexact matching of those representations which can be accomplished using the *mismatch(k,m)*[1] kernel family [8]. Traditional mismatch kernels do not take into account any spatial information contained in k-mers which may be critical for accurate modeling of inexact relationships. More recently, Kuksa et al. introduced the *sparse*

[1] k denotes the k-mer length and $m < k$ is the maximal number of allowed mismatches.

spatial sample kernels (SSSK) [7] which model the intrinsic spatial information in substrings of X. In particular, they consider substrings of form $S = a_1 \overset{d_1}{\leftrightarrow} a_2, \overset{d_2}{\leftrightarrow}, \cdots, \overset{d_{t-1}}{\longleftrightarrow} a_t$ (a_1 separated by d_1 characters from a_2, a_2 separated by d_2 characters from a_3, etc.) and computationally efficiently compute their spectra and kernels. This representation leads to more accurate elucidation of remote homologies.

Nevertheless, inexact matchings are typically insufficient for ascertaining accurate relationship among remote homologs. A role in establishing those links can be fulfilled by unlabeled data. The unlabeled sequences serve as conduits for propagating information between distant homologs. The families of profile [6] kernels, the sequence neighborhood kernels [11] and the SSSK [7] achieve state-of-the-art performance by leveraging this source of information. All of the methods rely on the notion of sequence neighborhoods $N(X)$, the sets of sequences most similar to any particular sequence X. $N(X)$ is typically established using a scoring function $s(X, X')$, such as the e-value, yielding $N(X) = \{X' : s(X, X') \leq \delta\}$ for a fixed threshold δ. The profile kernels use this neighborhood information to construct probabilistic profiles that are subsequently used for inexact matching. The sequence neighborhood kernels, on the other hand, use the neighborhoods to smooth the sequence features $\Phi^{orig}(X)$,

$$\Phi^{new}(X) = \frac{1}{|N(X)|} \sum_{X' \in N(X)} \Phi^{orig}(X') \quad (1)$$

for each training and testing sequence. Weston et al. in [11] and Kuksa et al. in [7] show that the discriminative power of the classifiers improve significantly using this neighborhood information. However, as we show in the subsequent sections, the way $N(X)$ is constructed impacts in a large way the accuracy of the neighborhood methods.

3 Proposed methods

In Sec. 3.1, we first propose a new framework for extracting only relevant information from unlabeled data in a semi-supervised learning setting. We then extend the framework in Sec. 3.2 using clustering to reduce computational complexity and data redundancy, which, as we will show experimentally, further improves the speed and accuracy of resulting classifiers.

3.1 Extracting relevant information from the unlabeled sequence database

Under a semi-supervised learning setting, our goal is to recruit *neighbors* of training and testing sequences to construct $N(X)$ and use these intermediate neighbors to establish similarity between the remotely homologous proteins, which bear little to no similarity on the primary sequence level. As a result, the quality of the intermediate neighboring sequences is crucial for detecting remote homologues. However, in many sequence databases, multi-domain protein sequences are abundant and such sequences might be similar to several unrelated single-domain sequences, as noted in [11]. Direct use of these long sequences may falsely establish similarities among unrelated sequences since these unlabeled sequences carry *excessive* and unnecessary features. In contrast, very short sequences often induce very sparse representation and therefore have *missing* features. Explicit use of sequences that are too long or too short may bias the averaged neighborhood representation and compromise the classifier performance. A possible remedy is to discard neighboring sequences whose lengths are substantially different from the query (training or test) sequence. For example, Weston et al. in [11] proposed to only capture neighboring sequences with maximal length of 250 (for convergence purposes). Unfortunately, such practice may not offer a direct and meaningful biological interpretation. Moreover, removing neighboring sequences purely based on their length may discard those that carry crucial information and, as we will show in Sec. 4, degrade classification performance.

To more effectively use the unlabeled neighboring sequences, we propose to extract the *significantly similar sequence regions* from the unlabeled neighbors since these regions are more likely to be *biologically relevant*. Such significant regions are commonly reported in search methods such as BLAST [2], PSI-BLAST [1] and different HMM methods. We illustrate the proposed procedure using PSI-BLAST as an example in Figure 1. In the figure, given the query sequence, PSI-BLAST reports sequences (hits) containing substrings that exhibit statistically significant similarity with the query sequence. For each reported significant hit, we extract the most significant region and recruit the extracted sub-sequence as a neighbor of the query sequence. Thus, the region-based neighborhood $R(X)$ contains the *extracted significant sequence regions*, not the whole neighboring sequences of the query sequence X, *i.e.* $R(X) = \{x' : s(X, X') \leq \delta\}$, where $x' \sqsubseteq X'$ is the most statistically significant matching region of an unlabeled neighbor X'. The proposed region-based neighborhood method, as demonstrated in Sec. 4, will allow us to more efficiently leverage the unlabeled data and significantly improve the classifier performance.

We summarize four competing methods for leveraging unlabeled data during training and testing under the semi-supervised learning setting below and experimentally compare the methods in Sec. 4:

- *Unfiltered*: all neighboring sequences are recruited and $N(X)$ is established on the whole-sequence level.
- *Extracting the most significant region*: for each re-

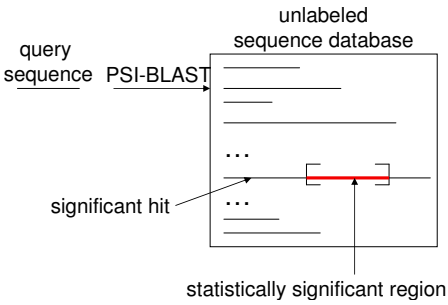


Figure 1. Extracting only statistically significant regions (red/light color) from the hits

cruited neighboring sequence, we extract only the most *significantly similar sequence region* and establish the region-based neighborhood $R(X)$ on a sub-sequence level; such sub-sequence is more likely to be biologically relevant to the query.

- *Filter out long and short sequences*: for each unfiltered neighborhood $N(X)$, we remove all sequences $X' \in N(X)$ if $T_{X'} > 2T_X$ or $T_{X'} < \frac{T_X}{2}$, where T_X is the length of sequence X. In essence, this method may alleviate the effect of the excessive and missing features in the unfiltered method by discarding the sequences whose length fall on the tails of the histogram.
- *Maximal length of 250* [11]: for each sequence, we first construct $N(X)$, then remove all neighboring sequences $X' \in N(X)$ if $T_{X'} > 250$.

3.2 Clustered Neighborhood Kernels

The smoothing operation in Eq. 1 is susceptible to overly represented neighbors in the unlabeled data set since the presence of replicates in $N(X)$ biases the average towards such sequences. Large uncurated sequence databases usually contain abundant duplicate sequences. Some duplicates, such as those with *secondary accession numbers* in Swiss-Prot, can be easily identified and removed. However, two other types of duplication are harder to identify: the sequences that are nearly identical and the sequences that contain substrings sharing high sequence similarity and are significant hits to the query sequence. Pre-processing the data prior to kernel computations is thus necessary to remove such bias and improve performance.

In this study we propose the approach of *clustered neighborhood kernels*. Clustered neighborhood kernels further simplify $R(X)$ to obtain a reduced region neighborhood $R^*(X) \subseteq R(X)$ without duplicate or near-duplicate regions (*i.e.* no pair of sequence regions in $R^*(X)$ shares more than a pre-defined sequence identity level). The simplification is accomplished by clustering the set $R(X)$. The clustered region-based neighborhood kernel between two sequences is then

$$K'(X,Y) = \sum_{x \in R^*(X)} \sum_{y \in R^*(Y)} \frac{K(x,y)}{|R^*(X)||R^*(Y)|}. \quad (2)$$

Clustering typically incurs quadratic complexity in the number of sequences [2, 9]. Moreover, *pre-clustering* the unlabeled database may result in the loss of neighboring sequences, further degrading the classifier performance, see Sec. 5.2. To address the two issues we propose to *post-cluster* each reported neighbor set *one at a time*. For example, the union of all neighbor sets induced by the NR unlabeled database contains $129,646$ sequences, while the average size of the neighbor sets is only 115.

4 Experiments

We present experimental results for protein remote homology detection under the semi-supervised setting on the SCOP 1.59 [10] benchmark data set [11]. The data set contains 54 binary classification problems, each simulating the remote homology detection problem by completely holding out a whole family for testing the super-family classifier. We use three unlabeled sequence databases, some containing abundant multi-domain protein sequences and duplicated or overly represented (sub-)sequences: PDB [3][2] (17,232 sequences), Swiss-Prot[3] (101,602 sequences), and the *non-redundant* (NR) sequence database (534,936 sequences). To adhere to the true semi-supervised setting, we remove *all sequences in the unlabeled data sets identical to any test sequences*.

The unfiltered $N(X)$ is constructed using two PSI-BLAST iterations on the unlabeled database with query X and the selection threshold based on e-values $\leq .05$. Next for each neighboring sequence, we extract the most significant region (lowest e-value) to form the sub-sequence neighborhood $R(X)$. Finally, we cluster $R(X)$ at 70% sequence identity level using *cd-hit*[4] [9], and form the *clustered neighborhood* $R^*(X)$. The neighborhood kernel is then obtained using the smoothed representations (Eq. 1) by substituting $N(X)$ with $R(X)$ or $R^*(X)$.

We evaluate all methods using the *Receiver Operating Characteristic* (ROC) and ROC50 [4] scores. In all experiments, we normalize the kernel values $K(X,Y)$[5] and use an existing SVM implementation SPIDER[6] with default parameters. For the sparse spatial sample kernel, we use the triple(1,3) (k=1, t=3, d=3), *i.e.* features are *triples* of monomers, and for the mismatch kernel, we use k=5, m=1. More details and supplementary data can be found at `http://seqam.rutgers.edu/projects/bioinfo/region-semiprot/`.

[2] As of Dec. 2007.
[3] Version used in [11] for comparative analysis of performance.
[4] http://www.cd-hit.org
[5] $K'(X,Y) = K(X,Y)/\sqrt{K(X,X)K(Y,Y)}$.
[6] http://www.kyb.tuebingen.mpg.de/bs/people/spider

4.1 Experiments with triple(1,3)

In the upper panel of Figure 2, we show the ROC50 plots of all four competing methods, *with post-clustering*, using the triple(1,3) kernel on different unlabeled sequence databases. In each figure, the horizontal axis corresponds to a ROC50 score, and the vertical axis denotes the number of experiments, out of 54, with equal or higher ROC50 score. In all cases, we observe the ROC50 curves of the region-based method (lines with '+' signs) show strong dominance over those of other methods. Furthermore, we observe in Figures 2(a) and 2(b), discarding sequences based on the sequence length (the two colored dashed lines) degrades the performance of the classifiers, compared to the baseline (unfiltered) method (solid lines). This suggests that longer unlabeled sequences carrying crucial information for inferring the class labels of the test sequences are discarded.

We also summarize performance measures for all competing methods in Table 1. For each method, we also report the p-value of the Wilcoxon Signed-Rank test on the ROC50 scores against the *unfiltered* (baseline) method. Our region-based method strongly outperforms other competing methods and consistently shows statistically significant improvements while the other two methods suggest no strong evidence of improvement. We also note that clustering significantly improves the performance of the unfiltered method (p-value $< .05$ in all unlabeled datasets) and offers noticeable improvements for the region-based method on larger datasets (*e.g.* NR).

Table 1. Experimental results for all competing methods using the triple(1,3) kernel.

	neighborhood			clustered neighborhood		
dataset	ROC	ROC50	p-value	ROC	ROC50	p-value
PDB						
unfiltered	.9476	.7582	-	.9515	.7633	-
region	.9708	**.8265**	**.0069**	.9716	**.8246**	**.0045**
no tails	.9443	.7522	.5401	.9472	.7559	.5324
max length	.9471	.7497	.4407	.9536	.7584	.5468
Swiss-Prot						
unfiltered	.9245	.6908	-	.9464	.7474	-
region	.9752	**.8556**	**2.46e-04**	.9732	**.8605**	**1.5e-03**
no tails	.9361	.6938	.8621	.9395	.7160	.6259
max length	.9300	.6514	.2589	.9348	.6817	.1369
NR						
unfiltered	.9419	.7328	-	.9556	.7566	-
region	.9824	**.8861**	**1.08e-05**	.9861	**.8944**	**2.2e-05**
no tails	.9575	.7438	.6640	.9602	.7486	.8507
max length	.9513	.7401	.8656	.9528	.7595	.8696

4.2 Experiments with mismatch(5,1)

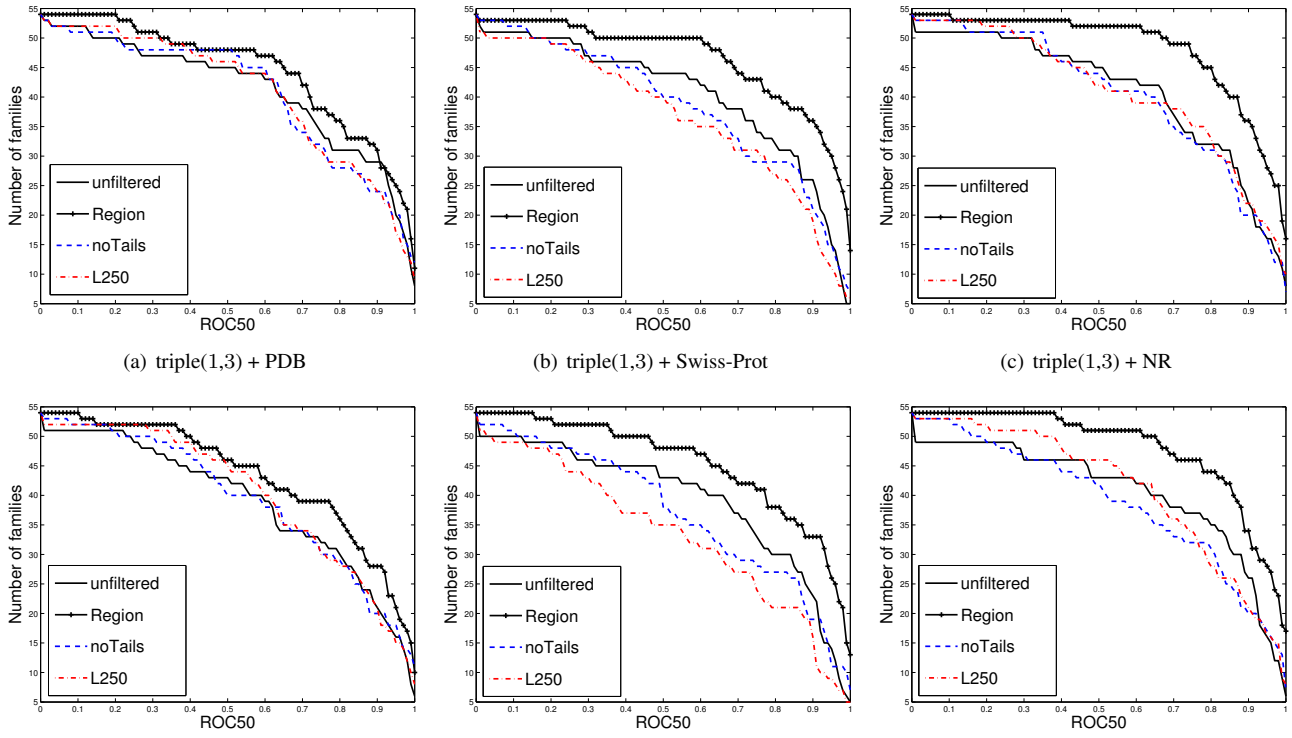
In the lower panel of Figure 2, we show the ROC plots of all four competing methods, *with post-clustering*, using the mismatch(5,1) kernel on different unlabeled sequence databases. We observe that the ROC50 curves of the region-based method show strong dominance over those of other competing methods. In Figures 2(e) and 2(f), we again observe the effect of filtering out unlabeled sequences based on the sequence length: longer unlabeled sequences carrying crucial information for inferring the label of the test sequences are discarded and therefore the performance of the classifiers is compromised. We summarize performance measures on all competing methods in Table 2. The region-based method again shows statistically significant improvement compared to the unfiltered and other methods. Similar to the triple kernel, we also observe significant improvements for the *unfiltered* method with clustered neighborhood on larger datasets.

Table 2. Experimental results on all competing methods using the mismatch(5,1) kernel.

	neighborhood			clustered neighborhood		
dataset	ROC	ROC50	p-value	ROC	ROC50	p-value
PDB						
unfiltered	.9389	.7203	-	.9414	.7230	-
region	.9698	**.8048**	**.0075**	.9705	**.8038**	**.0020**
no tails	.9379	.7287	.9390	.9378	.7301	.7605
max length	.9457	.7359	.4725	.9526	.7491	.3817
Swiss-Prot						
unfiltered	.9253	.6685	-	.9378	.7258	-
region	.9757	**.8280**	**.0060**	.9773	**.8414**	**.0108**
no tails	.9290	.6750	.9813	.9344	.6874	.5600
max length	.9185	.6094	.1436	.9223	.6201	**.0279**
NR						
unfiltered	.9475	.7233	-	.9544	.7510	-
region	.9837	**.8824**	**1.7e-04**	.9874	**.8885**	**1.2e-04**
no tails	.9554	.7083	.7930	.9584	.7211	.7501
max length	.9508	.7421	.7578	.9518	.7613	.9387

4.3 Comparison with other state-of-the-art methods

In Table 3, we compare our proposed methods on two string kernels (triple and mismatch) against the profile kernel, the state-of-the-art method. For each unlabeled data set, we highlight the methods with the best ROC50 scores. In almost all cases, the region-based method with clustered neighborhood demonstrates the best performance. Moreover, the ROC50 scores of the triple and mismatch kernels using regions strongly outperform those of the profile kernel. We note that previous studies [6, 11] suggest that the profile kernel outperforms the mismatch neighborhood kernel. Moreover, as shown in [6], to improve the accuracy of the profile kernels, one needs to increase the computationally demanding PSI-BLAST iterations. Using the



(a) triple(1,3) + PDB (b) triple(1,3) + Swiss-Prot (c) triple(1,3) + NR

(d) mismatch(5,1) + PDB (e) mismatch(5,1) + Swiss-Prot (f) mismatch(5,1) + NR

Figure 2. ROC50 plots of four competing methods using the triple-(1,3) (top) and mismatch-(5,1) (bottom) kernels with PDB, Swiss-Prot and NR as unlabeled databases. The ROC50 curves of the region-based neighborhood method consistently show strong dominance over all competing methods.

region-based neighborhood with only 2 PSI-BLAST iterations both the mismatch and spatial neighborhood kernels achieve results better than the best profile kernels with 5 PSI-BLAST iterations [6].

Table 3. Comparison of performance (ROC50) against the state-of-the-art methods.

method	PDB	Swiss-Prot	NR
triple(1,3)	.7582	.6908	.7327
triple(1,3), region	**.8265**	.8556	.8861
triple(1,3), region, clustering	.8246	**.8605**	**.8944**
mismatch(5,1)	.7203	.6685	.7233
mismatch(5,1), region	.8048	.8280	.8824
mismatch(5,1), region, clustering	.8038	.8414	.8885
profile(5,7.5)	.7205	.7914	.8151

5 Discussion

We further discuss the benefits of extracting only statistically significant regions from the neighboring sequences in Sec. 5.1 and elaborate on the role of post-clustering in Sec. 5.2

5.1 Motivation for region extraction

Figure 3 illustrates the benefit of extracting only statistically significant regions from the unlabeled sequences. In the figure, colors indicate membership: yellow (shaded) represents the positive class and green (pattern) the negative class. The arcs indicate (possibly weak) similarity induced by shared features (black boxes) and absence of arcs indicates no similarity. Sequences sharing statistically significant similarity are more likely to be evolutionarily related and therefore to belong to the same superfamily. As can be seen from the figure, the positive training and test sequences share no features and therefore no similarity; however, the unlabeled sequence shares some features with both sequences in the reported region, which is very likely to be biologically related to both positive sequences. Via this unlabeled sequence, the similarity between the two positive sequences is established. In contrast, if the whole unlabeled sequence is recruited as a neighbor, the similarity between the positive training and negative test sequences will be falsely established by the irrelevant regions, resulting in poor classifier performance.

One example in the SCOP 1.59 dataset that demon-

strates this behavior is the target family *EGF-type module* under the *EGF/Laminin* superfamily, *Knottins* fold and *small proteins* class. In the experiment, we observe an unlabeled sequence in Swiss-Prot (ID Q62059) sharing statistically significant similarity to the positive training, positive test, and negative test sequences. Swiss-Prot annotation states that this protein sequences contain the *C-type lectin*, *Immunoglobulin-like V-type*, *link* and *sushi (CCP/SCR)* domains. Without region extraction, the ROC50 scores are 0.3250 and 0.3292 under the triple and mismatch kernels. By establishing the neighborhood based on the extracted regions, the ROC50 scores improve to 0.9464 and 0.9664.

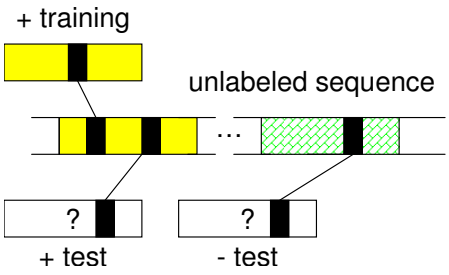


Figure 3. The importance of only extracting relevant region from neighboring sequences (middle) for inferring sequence labels (see text for more details).

5.2 The role of clustered neighborhoods

In Sec. 3.2, we propose to *post-cluster* each sequences neighbor set *one at a time*, as opposed to *pre-clustering* the union of all neighbor sets or the whole unlabeled sequence database. In this section, we further illustrate the benefits of post-clustering: improvement in performance as well as reduced storage and running time for classification.

We first show the difference between pre- and post-clustering using the PDB database. For the triple(1,3) neighborhood kernel, the ROC-50 scores for pre-/post- clustering are .8122 and .8246 with a border-line significant p-value of .1248. For the mismatch(5,1) kernel, the ROC-50 scores for pre-/post- clustering are .7836 and .8038 with a significant p-value of .0853. Potentially useful neighbors shared by two sequences might be removed during pre-clustering and not included in the neighbor set which can result in worse performance compared to post-clustering.

In addition to improving classification accuracy, performing clustering on the neighbor sets may also lead to substantial reduction in computational time. *Post-clustering* on the non-redundant sequence database takes approximately 120 seconds and on average reduces the neighborhood size in half (on NR, the average neighborhood sizes are $|N(X)|$=$|R(X)|$=115 *without clustering* and $|R^*(X)|$=67 *with clustering*). We also observe that our proposed framework reduces the running time by three folds.

6 Conclusion

We propose a systematic and biologically motivated approach for extracting relevant information from unlabeled sequence database to more efficiently leverage the power of the unlabeled data under the semi-supervised learning setting. We also propose the use of the clustered neighborhood kernels to improve the classifier performance and remove the kernel estimation bias caused by overly-represented sequences in large uncurated databases. Combined with two state-of-the-art string kernels (spatial and mismatch), our framework significantly improves accuracy and achieves the state-of-the-art performance on semi-supervised protein remote homology detection while exhibiting significant improvement in running time. Our approach can be readily extended to other challenging sequence analysis tasks, such as fold prediction, clustering and localization.

References

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

[2] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. Basic Local Alignment Search Tool. *Journal of Molecular Biology*, pages 403–410, 1990.

[3] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28:235–242, 2000.

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

[5] T. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1-2):95–114, 2000.

[6] 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. *J Bioinform Comput Biol*, 3(3):527–550, June 2005.

[7] P. Kuksa, P.-H. Huang, and V. Pavlovic. Fast protein homology and fold detection with sparse spatial sample kernels. In *Proceedings of the Nineteenth International Conference on Pattern Recognition (ICPR 2008)*, 2008.

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

[9] W. Li and A. Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, 2006.

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

[11] 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.