Community Detection-based Feature Construction for Protein Sequence Classification

Karthik Tangirala, Nic Herndon, and Doina Caragea

Department of Computing and Information Sciences
Kansas State University, Manhattan, KS 66502
{karthikt,nherndon,dcaragea}@ksu.edu

Abstract. Machine learning algorithms are widely used to annotate biological sequences. Low-dimensional informative feature vectors can be crucial for the performance of the algorithms. In prior work, we have proposed the use of a community detection approach to construct low dimensional feature sets for nucleotide sequence classification. Our approach uses the Hamming distance between short nucleotide subsequences, called k-mers, to construct a network, and subsequently uses community detection to identify groups of k-mers that appear frequently in a set of sequences. While this approach worked well for nucleotide sequence classification, it could not be directly used for protein sequences, as the Hamming distance is not a good measure for comparing short protein k-mers. To address this limitation, we extend our prior approach by replacing the Hamming distance with substitution scores. Experimental results in different learning scenarios show that the features generated with the new approach are more informative than k-mers.

Keywords: community detection, feature construction, feature selection, dimensionality reduction, protein sequence classification, supervised learning, semi-supervised learning, domain adaptation

1 Introduction

Machine learning has been extensively used to address prediction and classification problems in the field of bioinformatics. Advancements in sequencing technologies have led to the availability of large amounts of sequential data (mostly unlabeled), which can benefit learning algorithms. In general, most learning algorithms require a vectorial representation of the data in terms of features. Representing the data through low-dimensional informative feature sets is critical for the performance of the algorithms, in terms of both accuracy and complexity.

However, for many biological problems it is not yet understood which features are informative. In the absence of known informative features, it is common to represent the sequences as the count of k-mers generated using a sliding window-based approach. To do this, a window of a particular size, k, is traversed across the sequence, and at each step in the traversal, the fragment of the sequence within the window is captured. All such possible unique subsequences/fragments (referred to as k-mers) are used as features to represent sequences. As informative features can have variable length, the size, k, of the window is varied. However, variable length k-mers result in high-dimensional feature sets, increased computational complexity and sometimes decreased classification accuracy.

Feature selection is one of the techniques widely used to reduce the dimensionality of the input feature space, while retaining most of the informative features. Most of the feature selection techniques use the available labeled data to estimate feature-class dependency scores for all features. The features are then filtered based on the corresponding feature-class dependency scores. In theory, feature selection can be applied not only in supervised learning (large amounts of labeled data is used in the learning process), but also in semi-supervised learning (small amounts of labeled and large amounts of unlabeled data are used) and domain adaptation (large amounts of labeled data from a source domain, along with small amounts of labeled data and large amounts of unlabeled data from a target domain are used to learn classifiers for the target data). However, in the semi-supervised and domain adaptation, as the amount of available (target) labeled data is small, feature selection may not capture the feature-class dependencies accurately. Furthermore, when the number of features is very large, feature selection techniques might be computationally expensive. Therefore, alternative methods to generate a reduced set of informative features can presumably benefit supervised, semi-supervised and domain adaptation algorithms.

Towards this goal, in [16], we have introduced the idea of using a community detection algorithm to generate a low-dimensional informative sequential feature set for classifying nucleotide sequences (specifically, for the problem of classifying exons as either alternatively spliced or constitutive). Our approach extended TFBSGroup [15], an unsupervised approach to identify transcription factor binding sites in a small number of nucleotide sequences, based, in turn, on the community detection algorithm proposed in [23]. The worst case running time of TFBSGroup is quartic in the total number of sequences and the length of each sequence in the dataset. As a result, running TFBSGroup on large sets of sequences has high computational cost. We proposed a fast and novel extension to TFBSGroup [16], which makes it possible to generate features for large sets of nucleotide sequences. Our approach is based on randomly sampling small subsets of sequences (as opposed to using all the sequences at once) and finding informative features in each set separately. The final set of informative features is obtained by taking the union of the individual sets found using TFBSGroup. Although our prior approach [16] was successfully used to identify low-dimensional informative features (referred to as c-mers) for nucleotide sequences, it cannot directly be applied for protein sequences given the large size of the protein alphabet and the short length of the informative protein k-mers.

To address this limitation, in this paper, we further extend the approach in [16] to protein sequences by making use of amino acid substitution scores in place of the Hamming distance, under the assumption that the substitution scores are better than the Hamming distance when comparing short protein subsequences. We have applied the proposed approach to the problem of classifying protein sequences based on their localization. To evaluate the predictive power of c-mers in classifying protein sequences, we have conducted experiments in three different learning scenarios: supervised, semi-supervised and domain adaptation. Experimental results in all three learning scenarios suggest that the features generated with the community detection approach are more informative than k-mers in classifying protein sequences.

The rest of the paper is organized as follows: The related work on applications that have used k-mers, feature selection and community detection approaches is described in Section 2. The proposed approach of using a community detection algorithm to generate features for biological sequences is discussed in Section 3. Section 4 lists the research questions that we are addressing through this work, along with details about the set of experiments conducted, and the datasets used. The results of the experiments conducted are presented in Section 5, followed by conclusions in Section 6.

2 Related Work

In bioinformatics, and especially biological sequence classification, the sliding window approach is frequently used, sometimes together with feature selection or a different dimensionality reduction method, to generate k-mers and represent biological sequences as vectors of k-mers [1–3]. As an alternative to feature selection, we propose to use community detection to select a small set of informative features (specifically, k-mers that appear frequently in a set of sequences).

To find communities, Grivan and Newman [8, 22] proposed a hierarchical divisive algorithm, that iteratively removes edges between nodes based on their "betweenness", until the modularity of a partition reaches the maximum. The "betweenness" measure defines the total number of shortest paths between any two nodes that pass through an edge. The authors estimated the modularity of a partition, referred to as the Newman-Girvan modularity, by comparison with a null model (random graph). Their algorithm is believed to be the first of modern day community detection approaches. Clauset et al. [4] proposed a fast community detection approach that uses the Newman-Girvan modularity gain. Their approach starts with a set of isolated nodes, and the nodes are iteratively grouped based on the modularity gain. While some techniques use exhaustive optimization to better estimate the final maximum modularity, at the expense of computational cost [9–12], more efficient techniques have also been proposed to identify communities from large complex networks [23–27].

In bioinformatics, community detection has been mainly used in the context of protein-protein interaction networks and prediction of functional families [18–21]. Jia et al. [15] used community detection to identify transcription factor binding sites in a small set of nucleotide sequences (approach referred to as TFBSGroup). In [16], we have extended TFBSGroup to construct sequential features for classifying large sets of nucleotide sequences. To the best of our knowledge, community detection algorithms have not been used to construct sequential features for classifying protein sequences in a machine learning framework.

3 Feature Construction Using Community Detection

3.1 Community Detection Algorithm

Complex network analysis has gained a lot of attention among researchers interested in identifying hidden structural and relational properties within a large system. A network, similar to a graph, comprises of a set of V nodes, $\{n_1, n_2, \cdots, n_V\}$, along with a set of E edges, $\{(n_i, n_j) \mid 1 \leq i \neq j \leq V\}$. Many complex systems can be represented using a network, with nodes being the elementary components of the system and the relationship between the components being the links.

A community is a sub-network whose nodes are highly connected with each other, as compared to other nodes outside the community. Thus, a community reflects a group of closely related nodes. Identifying communities can uncover structural properties of a network. From the methods available to identify communities, we use a technique based on modularity, proposed by Blondel et al. [23].

The modularity of a network (denoted by Q) measures the structure of a network by defining the strength of the network when divided into modules (sub-networks or communities). High modularity suggests that the nodes within each community are densely connected when compared to other nodes. The algorithm proposed in [23] identifies communities by optimizing the modularity gain. It is a fast and efficient approach to identify high modularity partitions in a large network, which can be seen as a two-phase iterative process.

In the first phase, each node is assigned to a different community. Then, for each node, n_i, the algorithm computes the gain in modularity, ΔQ, achieved by removing n_i from its community and placing it in the community of n_j, where n_j is a neighboring node of n_i. It then assigns n_i to the community of that n_j, for which the maximum modularity gain is obtained. In the second phase, a new network is constructed, with the nodes being the communities identified in the first phase. The weights of the edges between the new nodes are computed as the sum of the weights of the edges between nodes of the corresponding two communities. Edges among nodes of the same community form self-loops in the new network. These two phases are iterated until there is no further improvement in the modularity gain, and, then, the final set of communities is returned.

3.2 Identifying Motifs Using Community Detection

Jia et al. [15] introduced the idea of using community detection to identify transcription factor binding sites (a.k.a., motifs) in a set of nucleotide sequences. A motif is a pattern that is widespread across different sequences, and potentially has biological significance. Consequently, a motif can be obtained by aligning a set of subsequences that occur across different sequences (called motif instances), which are highly correlated to each other. The motif is also referred to as the consensus of its motif instances. The approach proposed by Jia et al. [15], called TFBSGroup, aims at identifying motifs under the ZOMOPS constraint (Zero, One or Multiple Occurrences of a motif Per Sequence). The motifs identified have length k, and there are at most d mismatches between motif instances and the motif consensus. For a set of N sequences of maximum length L, the TFBSGroup approach also works in three phases/steps.

(Step 1) The first step deals with the construction of an N-partite network and detection of communities in that network. The nodes of the network represent all possible k-mers (subsequences of length k) of the input sequences. Therefore, for a set of N sequences, each of length L, there are $(N * (L - k + 1))$ nodes. Two nodes are connected by an edge only if the Hamming distance between the k-mers corresponding to the two nodes is no more than x (a parameter that the TFBSGroup algorithms takes). Given that the maximum Hamming distance allowed between a motif instance and the motif consensus is d (another TFBSGroup parameter), it follows that the maximum Hamming distance between any two motif instances is $2d$. Therefore, while constructing the network, the maximum value that x can be given is $2d$. We should note that there is no edge between nodes (k-mers) belonging to the same sequence, which means that a set of N sequences results in an N-partite network.

(Step 2) After constructing the network, all possible communities of size at least q (another parameter) are identified. Then, from each community, a motif consensus is generated by aligning all k-mers from that particular community.

(Step 3) Finally, each motif consensus is greedily refined towards a final motif, and a significance score is calculated for it. The top t motifs (default t=10) are then selected based on the significance score (see [15] for more details).

According to [15], the worst-case time complexity of the TFBSGroup algorithm is quartic in terms of the total number of input sequences, N, and the length of the sequences, L: $O(p(k, x)^2 \times N^4 \times L^4)$, where $p(k, x)$ is the probability of two random k-mers having Hamming distance at most x. Although TFBSGroup can successfully identify transcription factor binding sites in a small set of sequences, it cannot be applied for generating features for classification problems, due to the large number of sequences involved. To address this problem, in [16], we proposed an approach for scaling up TFBSGroup, as described below.

3.3 Feature Construction for Large Nucleotide Sequence Datasets

To extend TFBSGroup to generate features for sequence classification problems, in [16], we proposed to run TFBSGroup on a set of randomly selected R samples, each of S sequences, from the available data consisting of N sequences, where $S \ll N$. The time complexity of running TFBSGroup on R *samples* reduced to:

$O(p(k, x)^2 \times S^4 \times L^4 \times R) \ll O(p(k, x)^2 \times N^4 \times L^4)$, when $(R \times S^4) \ll N^4$. We choose R and S that satisfy the condition above to achieve scalability. Furthermore, when generating the R samples, we allow overlap between samples, but there is no overlap between sequences within a sample. The reason for this is that we are interested in finding patterns/motifs that are frequent across sequences, but not necessarily within a sequence. By allowing samples to overlap, we can essentially link subsequences in different samples, and get higher coverage.

We run TFBSGroup on each individual sample and select the top t motifs from each sample. All the resulting motifs are merged together to form the final set of motifs. As a result, the final set of motifs contains a total of $t \times R$ motifs (for a particular length of the motif, k). The frequency count representation of all unique motif instances present in the final set of motifs is then used to represent sequences for classification. We refer to the set of motif instances as the set of c-mers given that they are identified based on community detection. This approach has been successfully used to construct informative features for learning classifiers from large sets of nucleotide sequences [16]. However, it cannot be directly used to construct features for protein sequences, as the Hamming distance does not capture well differences between short protein k-mers.

3.4 Feature Construction for Large Protein Sequence Datasets

For protein sequences, motifs of shorter length carry better information than motifs of longer length [2]. When the length of the motif is small (*e.g.*, $k = 1$, 2 or 3), the probability of two protein k-mers having Hamming distance less than a particular threshold, x, is high, as $x \approx k$, and thereby the resulting network in very dense. For longer k-mers (*e.g.*, $k = 6$, 7 or 8), usually the desired threshold x is smaller than k. In such cases, given the large alphabet size of protein sequences, the probability of having an edge between two nodes is very low, thereby resulting in a very sparse network. Therefore, when Hamming distance is used to construct a network of protein subsequences, the resulting network is either too sparse or too dense. To address this issue, we propose to use substitution scores when constructing sequential features for protein sequences. Substitution scores are computed using substitution matrices for amino acids, which take into account the divergence time as well as the substitution rate for each possible alignment of amino acids. Based on the default parameters for BLAST, in this work, we used PAM30 matrix [5, 13] to compute the substitution scores for pairs of k-mers.

Similar to the Hamming distance, the substitution score for a pair of k-mers is computed based on the alignment of the amino acids at the respective positions of the k-mers. However, when using substitution matrices, as opposed to the Hamming distance, the score of an alignment at a particular position is affected not only by the match/mismatch of the respective amino acids, but also by the degree of match/mismatch as captured by the substitution matrix. For example, consider two pairs of *3*-mers: $\{JQK, LZK\}$ and $\{PGD, RGD\}$. For pair 1, the Hamming distance is 2 and the substitution score is 19, and for pair 2, the Hamming distance is 1 and the substitution score is 10 (where substitution scores are computed using PAM30 matrix [5]). We should note that the substitution scores represent similarity scores, as opposed to distances. The higher the substitution score values, the more similar the sequences. Thus, based on the Hamming distance, the k-mers of pair 2 are more similar than the k-mers of pair 1. Contrarily, based on substitution scores, the k-mers of pair 1 are more similar when compared to pair 2. Given the fact that substitution scores capture the degree of match/mismatch, they are preferable to the Hamming distance when interested in identifying similar protein sequences.

In this work, we find protein motifs with the property that the substitution score between any motif instance and the corresponding motif consensus is at least s (a parameter of the algorithm). When constructing the subsequence network, a pair of nodes (protein subsequences of length k) are connected by an edge only if the substitution score of the two k-mers is greater than a particular threshold p (to avoid spurious edges, in our experiments, we chose $p = s/2$). After constructing the network, all possible communities of size at least q are identified in the network using the community detection algorithm, and the k-mers corresponding to each community are aligned to form the motif consensus. Subsequently, each motif consensus is greedily refined towards the final motif for that community, using the substitution scores, and, for each community, the refined motif along with the motif instances are returned, together with a normalized substitution score. The above process is repeated for all communities identified by the algorithm. The top t motifs (default t=10) from all the resulting motifs are then selected based on the normalized substitution scores. The unique motif instances belonging to the final motifs are used as classification features and will be denoted by c-mers, similar to how the community detection-based features for nucleotide sequences are denoted in our previous work.

4 Experimental Setup

In Section 4.1, we present the research questions addressed through our work. The datasets used are described in Section 4.2, and the details about the experimental setup are presented in Section 4.3.

4.1 Research Questions

We addressed the following research questions:

1. *How does the number of* c*-mers compare to the number of all possible* k*-mers?* The set of c-mers generated using the community detection algorithm satisfy the ZOMOPS constraint (Zero, One or Multiple Occurrences of the motifs Per Sequence). Therefore, we expect the dimensionality of the set of c-mers to be very small when compared to the dimensionality of the set of all k-mers.
2. *How does the predictive power of* c*-mers compare to that of* k*-mers?* To investigate the predictive power of c-mers, we compare the performance of the classifiers learned from sequences represented using c-mers with that of the classifiers learned from sequences represented using an equal number of k-mers (obtained via feature selection from the total number of k-mers). Given that the community detection approach used to generate c-mers is not supervised (i.e., does not make use of sequence labels), we have conducted experiments in three different learning scenarios: supervised, semi-supervised and domain adaptation. While larger amounts of data are possibly available in the supervised learning scenario, the assumption in the semi-supervised and domain adaptation scenarios is that only small amounts of labeled data are available for the domain of interest. Therefore, we expect the features obtained using community detection to be more informative than the k-mers, at least in these scenarios, as feature-class dependencies may not be well captured by feature selection when the amount of labeled data is limited.

4.2 Datasets

In this work, we targeted the problem of classifying protein sequences based on their respective localization. We used four different protein sequence datasets:

- **PSORTb datasets [7]:** The Gram-negative (GN) dataset consists of 1444 sequences belonging to one of the five classes: cytoplasm (278), cytoplasmic membrane (309), periplasm (276), outer membrane (391) and extracellular (190). The Gram-positive (GP) dataset consists of 541 sequences belonging to one of the four classes: cytoplasm (194), cytoplasmic membrane (103), cellwall (61) and extracellular (183).
- **TargetP datasets [6]:** The plant (P) dataset consists of 940 sequences belonging to one of the four classes: chloroplast (141), mitochondrial (368), secretory pathway/signal peptide (269) and other (consisting of 54 proteins labeled nuclear and 108 proteins labeled cytosolic). The non-plant (NP) dataset consists of 2738 sequences belonging to one of the three classes: mitochondrial (361), secretory pathway/signal peptide (715) and other (consisting of 1224 proteins labeled nuclear and 438 proteins labeled cytosolic).

4.3 Experiments

As mentioned above, we conducted experiments in three different scenarios: supervised, semi-supervised, and domain adaptation. We used the naïve Bayes multinomial (NBM) classifier for the supervised scenario; the co-training iterative-based algorithm, with NBM as the base classifier, for the semi-supervised scenario; and the algorithm proposed in [14], derived from the NBM classifier, for the domain adaptation scenario. All experiments are conducted using 5-fold cross-validation, with four folds used for training and the remaining fold for testing. In the supervised scenario, all the training data was assumed to be labeled and was used to learn the classifiers. In the semi-supervised scenario, we assumed 20% of the training data to be labeled and up to 80% to be unlabeled (specifically, we experimented with 20%, 40%, 60%, and 80% of the training data as unlabeled). Finally, for the domain adaptation scenario, we assumed a source domain with labeled data to be available, in addition to the target domain labeled and unlabeled data. We conducted experiments with the following pairs of source→target domains: $GP \rightarrow GN$, $GN \rightarrow GP$, $P \rightarrow NP$ and $NP \rightarrow P$, respectively. Only the overlapping classes within a pair of domains were used in the domain adaptation scenario (i.e., cytoplasm, cytoplasmic membrane and extracellular for the GP/GN pairs, and mitochondrial, secretory pathway/signal peptide and others for P/NP pairs). For each run, we used all the data from the source domain, and we split the training target data into 20% as labeled and up to 80% as unlabeled (i.e., 20%, 40%, 60%, and 80%).

We evaluated the performance of the classifiers using the area under the receiver operating characteristic curve (AUC), as the class distribution is relatively balanced for all data sets. We report the average AUC over the five runs.

Our goal is to compare two feature representations, c-mers and k-mers. The details of how these sets of features were generated are provided below:

- For c-mers, we invoke the proposed approach (Section 3.4) with the following parameter values: length of the motif $k \in \{2, 3, 4\}$, minimum community size $q = 5$, minimum substitution score $s = 15$, number of samples $R = 50$, sample size $S = 10$, and number of motifs selected $t = 10$. The algorithm returns the set of c-mers. We denote the number of c-mers by N_{c-mers}. We should note that the total number of unique sequences from all the R samples, N_{RS}, can be smaller than the total number of all the sequences in the dataset N, as the samples generated can have overlapping sequences.
- For k-mers, we use the sliding window approach. To make a fair comparison between c-mers and k-mers, we generate k-mers of the same length $k \in \{2, 3, 4\}$, on the same set of training sequences, N_{RS}. In addition, when comparing the performance, we apply feature selection on k-mers, using the labeled data only, to select top k-mers, such that the number of k-mers used, N_{k-mers}, is the same as the number of, c-mers, N_{c-mers}. For feature selection, we use *Entropy based Category Coverage Difference* (ECCD), proposed in [17], as this measure makes use of both the distribution of the sequences containing the features and the frequency of the occurrence of a feature value within the sequence, to compute the feature-class dependency scores.

5 Results

Following are the total number of features (#c-mers, #k-mers) generated, averaged over five folds for all four datasets: GP (1976, 74203), GN (1823, 83060), P (1684, 77091) and NP (1751, 102815). As can be seen, the total number of c-mers is much smaller than the total number of k-mers. As the set of c-mers represents a reduced set of k-mers (c-mers $\subset$ k-mers), this means that our proposed approach can be seen as a dimensionality reduction technique for k-mers.

Table 1 shows the AUC values for the supervised, semi-supervised, and domain adaptation scenarios, respectively. As can be seen, the AUC values are higher when using c-mers as compared to k-mers in all scenarios, for most experiments, specifically, in 3 out of 4 cases for the supervised scenario, and 15 out of 16 cases for the semi-supervised scenario as well as for the domain adaptation scenario. In semi-supervised/domain adaptation learning scenarios, given that the amount of available labeled data is small, feature selection may not estimate the feature-class dependencies accurately, thereby selecting a set of possibly uninformative features, while filtering out informative features. Thus, in cases when the amount of available labeled data is small, c-mers are expected to outperform k-mers selected using feature selection. Surprisingly, we observed a similar behavior also in the supervised learning scenario, the scenario where we presumably have sufficient labeled data to estimate the feature-class dependencies accurately. The reason for this might be that feature selection can still leave out informative features, as the size of the final set is limited by the number of c-mers, and it possibly includes some uninformative features. On the other hand, the set of c-mers, having the same size as the set of selected k-mers, capture much better features that carry information about classes.

Table 1: AUC values on the four datasets: Gram-positive (GP), Gram-negative (GN), plant (P), and non-plant (NP). For the semi-supervised and domain adaptation scenarios, the amount of labeled data is fixed to 20%, while the amount of unlabeled data is varied from 20% to 80%. For domain adaptation, the source and target domains are indicated as source→target. Note that for each scenario, in most cases, the classifiers had higher AUCs when using *c*-mers.

Supervised learning scenario								
Unlabeled	GP		GN		P		NP	
	c-mers	*k*-mers	*c*-mers	*k*-mers	*c*-mers	*k*-mers	*c*-mers	*k*-mers
0	**0.925**	0.869	**0.929**	0.915	**0.837**	0.754	0.834	**0.874**

Semi-supervised learning scenario								
Unlabeled	GP		GN		P		NP	
	c-mers	*k*-mers	*c*-mers	*k*-mers	*c*-mers	*k*-mers	*c*-mers	*k*-mers
20	**0.847**	0.746	**0.877**	0.793	**0.72**	0.663	**0.825**	0.787
40	**0.831**	0.748	**0.882**	0.793	**0.705**	0.656	**0.793**	0.788
60	**0.852**	0.742	**0.87**	0.783	**0.698**	0.655	0.773	**0.774**
80	**0.822**	0.749	**0.851**	0.788	**0.705**	0.659	**0.773**	0.77

Domain adaptation scenario								
Unlabeled	GN→GP		GP→GN		NP→P		P→NP	
	c-mers	*k*-mers	*c*-mers	*k*-mers	*c*-mers	*k*-mers	*c*-mers	*k*-mers
20	**0.877**	0.785	**0.911**	0.899	**0.802**	0.78	**0.829**	0.763
40	**0.856**	0.755	**0.91**	0.893	**0.748**	0.728	**0.821**	0.777
60	**0.852**	0.731	**0.902**	0.896	**0.739**	0.728	**0.777**	0.73
80	**0.839**	0.741	0.892	**0.895**	**0.734**	0.727	**0.803**	0.744

Together, the small dimensionality of the set of *c*-mers and the performance results in Table 1 suggest that our approach is successful in retaining informative/predictive features, while reducing the dimensionality by a large extent in all learning scenarios considered.

6 Conclusion

We have investigated the predictive power of the features generated using a community detection approach, for classifying proteins based on their respective localizations. As the original approach of using Hamming distance [15, 16] to generate nucleotide features does not work for sequences of a large alphabet size (such as proteins), we proposed a novel idea of using substitution scores as a similarity metric between two protein *k*-mers in the process of constructing the protein subsequence network. The resulting *c*-mers are associated with a set of motifs which represent groups of similar subsequences that occur frequently in the set of sequences. As opposed to that, the set of *k*-mers generated with a sliding window take into account all possible subsequences of a certain length occurring in the sequences. Both approaches are unsupervised, as they do not make use of class labels. To evaluate the predictive power of the features generated using our proposed approach (specifically, the predictive power of *c*-mers), we have conducted experiments in supervised, semi-supervised and domain adaptation learning scenarios. The results of the experiments show that our proposed approach generated low-dimensional informative features in supervised, semi-supervised and domain adaptation scenarios. Furthermore, those features have resulted in improved performance as opposed to *k*-mers selected based on feature-class dependency scores, even in the supervised scenario, where presumably there is enough labeled data to accurately estimate the scores.

Acknowledgments. This work was supported by an Institutional Development Award (IDeA) from the National Institute of General Medical Sciences of the National Institutes of Health under grant number P20GM103418. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute of General Medical Sciences or the National Institutes of Health. The computing for this project was performed on the Beocat Research Cluster at Kansas State University, which is funded in part by grants MRI-1126709, CC-NIE-1341026, MRI-1429316, CC-IIE-1440548.

References

1. C. S. Leslie, E. Eskin, A. Cohen, J. Weston, and W. S. Noble. Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4):467–476, 2004.
2. C. Caragea, A. Silvescu, and P. Mitra. Protein sequence classification using feature hashing. *Proteome Science*, 10(1):1–8, 2012.
3. L. Sun, H. Luo, D. Bu, G. Zhao, K. Yu, C. Zhang, Y. Liu, R. Chen, and Y. Zhao. Utilizing sequence intrinsic composition to classify protein-coding and long non-coding transcripts. *Nucleic Acids Research*, 2013.
4. A. Clauset, M. E. J. Newman, , and C. Moore. Finding community structure in very large networks. *Physical Review E*, pages 1– 6, 2004.
5. M. O. Dayhoff, R. M. Schwartz, and B. C. Orcutt. A model of evolutionary change in proteins. *Atlas of protein sequence and structure*, 5(suppl 3):345–351, 1978.
6. O. Emanuelsson, H. Nielsen, S. Brunak, and G. Heijne. Predicting subcellular localization of proteins based on their n-terminal amino acid sequence. *Journal of Molecular Biology*, 300(4):1005–1016, 2000.
7. J. L. Gardy, M. R. Laird, F. Chen, S. Rey, C. J. Walsh, M. Ester, and F. S. L. Brinkman. Psortb v.2.0: Expanded prediction of bacterial protein subcellular localization and insights gained from comparative proteome analysis. *Bioinformatics*, 21(5):617–623, 2005.
8. M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.
9. S.-P. M. A. L. Guimera, R. Modularity from fluctuations in random graphs and complex networks. *Phys. Rev. E*, 70(025101), 2004.
10. C. P. Massen and J. P. K. Doye. Identifying communities within energy landscapes. *Phys. Rev. E*, 71(046101), 2005.
11. A. Medus, G. Acuna, and C. Dorso. Detection of community structures in networks via global optimization. *Physica A: Statistical Mechanics and its Applications*, 358(2):593–604, 2005.

12. R. Guimera and L. A. N. Amaral. Functional cartography of complex metabolic networks. *Nature*, 433(7028):895–900, 2005.
13. S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992.
14. N. Herndon and D. Caragea. Naïve Bayes Domain Adaptation for Biological Sequences. In *Proceedings of the 4th International Conference on Bioinformatics Models, Methods and Algorithms*, BIOINFORMATICS 2013, pages 62–70, 2013.
15. C. Jia, M. Carson, and J. Yu. A fast weak motif-finding algorithm based on community detection in graphs. *BMC Bioinformatics*, 14(1):1–14, 2013.
16. K. Tangirala and D. Caragea. Community detection-based features for sequence classification. In Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (BCB '14). ACM, 2014.
17. C. Largeron, C. Moulin, and M. Gèry. Entropy based feature selection for text categorization. In *Proc. of the 2011 ACM Symp. on Applied Computing*, SAC '11, pages 924–928, 2011.
18. N. Dongfang and Z. Xiaolong. Prediction of hot regions in protein-protein interactions based on complex network and community detection. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 17–23, Dec 2013.
19. H. Mahmoud, F. Masulli, S. Rovetta, and G. Russo. Community detection in protein-protein interaction networks using spectral and graph approaches. In *Computational Intelligence Methods for Bioinformatics and Biostatistics*, Lecture Notes in Computer Science, pages 62–75. Springer International Publishing, 2014.
20. S. Mallek, I. Boukhris, and Z. Elouedi. Predicting proteins functional family: A graph-based similarity derived from community detection. In *Intelligent Systems'2014*, volume 323 of *Advances in Intelligent Systems and Computing*, pages 629–639. Springer International Publishing, 2015.
21. T. van Laarhoven and E. Marchiori. Robust community detection methods with resolution parameter for complex detection in protein protein interaction networks. In *Pattern Recognition in Bioinformatics*, volume 7632 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin Heidelberg, 2012.
22. M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(026113), 2004.
23. V. Blondel, J. Guillaume, R. Lambiotte, and E. Mech. Fast unfolding of communities in large networks. *J. Stat. Mech*, page P10008, 2008.
24. L. Donetti and M. A. Muñoz. Improved spectral algorithm for the detection of network communities. In *Proceedings of the 8th Granada Seminar - Computational and Statistical Physics*, pages 1–2, 2005.
25. U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3), Sept. 2007.
26. M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.
27. F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2658–2663, 2004.