Published in final edited form as:

IEEE Trans Nanobioscience. 2016 March ; 15(2): 84–92. doi:10.1109/TNB.2016.2523501.

A Comparative Analysis between k -mers and Community Detection-based Features for the Task of Protein Classification

Karthik Tangirala [Member, IEEE],

Computing and Information Sciences, Kansas State University, Manhattan, United States

Nic Herndon [Member, IEEE], and

Computing and Information Sciences, Kansas State University, Manhattan, United States

Doina Caragea [Member, IEEE]

Computing and Information Sciences, Kansas State University, Manhattan, United States

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 used the Hamming distance between short nucleotide subsequences, called k -mers, to construct a network, and subsequently used community detection to identify groups of k -mers that appear frequently in a set of sequences. Whereas 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 extended 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 classification; supervised learning; semi-supervised learning; domain adaptation

I. 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 accuracy and complexity of the algorithms.

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, with 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 that reduce the set of informative features can aid supervised, semisupervised and domain adaptation algorithms.

Towards this goal, in [1], we 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 [2], 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 [3]. 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. In [1] we proposed a fast and novel extension to TFBSGroup [1], which makes it possible to generate features for large sets of nucleotide sequences. Our approach was based on randomly sampling small subsets of sequences, as opposed to using all the sequences at once, and finding informative features in each subset separately. The final set of informative features is obtained by taking the union of the individual sets found using TFBSGroup. Although our prior approach was successfully used to identify lowdimensional informative features for nucleotide sequences, referred to as c -mers, 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, as further described in Section III–D.

To address this limitation, in this paper, we further extend the approach in [1] 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, as indicated in Section III–D. We applied the proposed approach to the problem of classifying protein sequences based on their localization. To evaluate the predictive power of pc -mers in classifying protein

sequences, we 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 II. The proposed approach of using a community detection algorithm to generate features for biological sequences is discussed in Section III. Section IV 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 V, followed by conclusions in Section VI.

II. 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 (e.g., [4], [5]). An alternative to feature selection is to identify a small set of informative motifs (e.g., [6]–[8]), specifically, k -mers that appear frequently in a set of biological sequences.

A large number of motif discovery approaches use probabilistic/statistical models to identify motifs, primarily represented in the form of position-specific scoring matrices (PSSM) [9], such as Multiple EM for Motif Elicitation (MEME) [10], Gibbs Sampler [11], and DRIMust [12]. Although these approaches can have a fast run time, attaining global optimum when identifying motifs with these approaches is sometimes difficult, especially when the motif length is small. Other approaches, such as MITRA-count [13], TFB-SGroup [2], and sMCL-WMR [14], typically search for (L, d) motifs based on the consensus model using various heuristic methods. In [15], we used Burrows Wheeler transform-based features to classify protein sequences, specifically for the problem of classifying proteins based on their localization.

Our proposed approach can be seen as a heuristic-based motif discovery approach, as it makes use of community detection to select a small number of informative features. In related work on community detection, Grivan [16] and Newman [17] 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. [18] 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. Whereas some techniques use exhaustive optimization to better estimate the final maximum modularity, at the expense of computational cost (e.g.,

[19], [20]), more efficient techniques have also been proposed to identify communities from large complex networks (e.g., [3]).

In bioinformatics, community detection has been mainly used in the context of protein-protein interaction networks and prediction of functional families (e.g., [21]–[23]). Jia et al. [2] used community detection to identify transcription factor binding sites in a small set of nucleotide sequences (approach referred to as TFBSGroup). In [1], we 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, with the exception of our own work in [24] that we are extending along several dimensions, as explained in Section III–D. Once we generate these features, they can be used with any classification algorithm that works with a feature vector representation. This includes naïve Bayes or naïve Bayes multinomial (using binary or motif count representation, respectively), support vector machines, logistic regression, etc. We focus on naïve Bayes multinomial because it is a simple classifier, has no parameters to tune, and we achieved good results with this classifier in our prior work with domain adaptation [15], [24], [25].

III. FEATURE CONSTRUCTION USING COMMUNITY DETECTION

A. 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, $\{n_1, n_2, \dots, n_V\}$, along with a set of E edges, $\{(n_i, n_j) \mid 1 \leq i, j \leq V\}$. Many complex systems can be represented using a network, with nodes as elementary components of the system, and the relationship between the components as 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. [3].

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 [3] 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 Q , 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 convergence of modularity gain, and then, the final set of communities is returned.

B. Identifying Motifs Using Community Detection

Jia et al. [2] 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. [2], 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 passed to the TFBSGroup algorithm. 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 [2] for more details).

According to [2], 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 [1], we proposed an approach for scaling up TFBSGroup, as described below.

C. Feature Construction for Large Nucleotide Sequence Datasets

To extend TFBSGroup to generate features for sequence classification problems, in [1] 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 nc -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 [1]. However, it cannot be directly used to construct features for protein sequences, as the Hamming distance does not capture well the differences between short protein k -mers.

D. Feature Construction for Large Protein Sequence Datasets

For protein sequences, short motifs (up to four amino acids long) carry better information than long motifs [4], [26], [27]. When the length of the motif is short (*e.g.*, $l = 1, 2$, or 3), the probability of two protein l -mers having Hamming distance less than a particular threshold, x , is high because $x \approx l$; therefore the resulting network is very dense. For long motifs (*e.g.*, $l = 6, 7$, or 8), the desired threshold x is typically smaller than l . However, given the large alphabet size of protein sequences, the probability of having an edge between two nodes is very low, resulting in a very sparse network. Thus, when the Hamming distance is used to construct a network of protein subsequences, the resulting network is either too sparse or too dense, making the use of our community detection approach for identifying motifs in nucleotide sequences impractical (as we end up with either too few or too many motifs). In order to address this problem, we propose a novel idea of using substitution scores in the process of constructing sequential features for protein sequences. Substitution scores are computed using PAM [28] substitution matrices for amino acids. PAM matrices are based on alignments of known homologous proteins, and take into account the divergence time as well as the substitution rate for each possible alignment of amino acids. Based on default parameters for BLAST, we used PAM30 substitution matrix to compute the substitution

score between two protein motifs as an attempt to capture similarities between short subsequences.

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 [28]). 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 are. 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 better 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 , which is 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 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 pc -mers, similar to how the community detection-based features for nucleotide sequences are denoted in our previous work.

IV. EXPERIMENTAL SETUP

In Section IV–A, we present the research questions addressed through our work. The datasets used are described in Section IV–B, and the details about the experimental setup are presented in Section IV–C.

A. Research Questions

We addressed the following research questions:

1. *How does the number of pc -mers compare to the number of all possible k -mers?*
The set of pc -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 pc -mers to be very small when compared to the dimensionality of the set of all k -mers. To investigate this property, the total number of pc -mers was compared to the total number of k -mers in a supervised learning scenario in which all available data was assumed to be labeled.
2. *How does the predictive power of pc -mers compare to that of k -mers?* To investigate the predictive power of pc -mers, we compare the performance of the classifiers learned from sequences represented using pc -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 pc -mers is not supervised, i.e., does not use the labels of sequence, we conducted experiments in three different learning scenarios: supervised, semi-supervised and domain adaptation. Whereas 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.
3. *How do results vary with amounts of unlabeled data?* Intuitively, for the same amount of labeled data for semi-supervised, or target labeled data for domain adaptation, increasing the amount of unlabeled data for semi-supervised, or target unlabeled data for domain adaptation, may result in increased performance; therefore, the amounts of unlabeled data was varied and variation in the performance of the classifiers in semi-supervised and domain adaptation learning scenarios was observed.
4. *How does the number of top motifs selected, t , affect the performance of the supervised learning classifier?* Each motif obtained is associated with a set of k -mers. The total number of features generated, the pc -mers, increases with the number of motifs selected from each of the R samples. Classifier performance, in general, increases with the number of features. However, the motifs are sorted based on substitution score. Because of this, as t increases, less significant motifs are added to the feature set, potentially adversely affecting classifier performance. Therefore, the performance of the classifier learned from top t motifs, where t is varied, should be investigated. Performance of the classifiers are expected to initially increase and then possibly decrease with an increase in the value of t .
5. *What is the effect of the number of samples, R , and sample size, S , on the performance of the classifier in a supervised learning scenario?* In supervised

learning scenario, what combination of R and S among (a) large R and large S , (b) relatively small R and small S , and (c) relatively small R and large S , yields the best results? Combination (b) is faster than the other two, and ideally it should give the best feature set.

6. *How does the threshold on the substitution score, s , affect the performance of the classifier in a supervised learning scenario?* Varying the threshold s affects the number of edges in the network, thereby affecting the communities obtained from the algorithm. Specifically, the use of extreme values of s creates degenerate networks, i.e., a very dense network for small s , or a sparse network for large s , potentially resulting in too many or too few features. In order to study this behavior, the threshold s was varied and the performance of the supervised learning classifier was recorded.

B. Datasets

In this work, we targeted the problem of classifying protein sequences based on their respective localization. We used four different protein sequence datasets (plant, non-plant, gram-positive bacteria, and gram-negative bacteria):

- **PSORTb datasets [29]:** The Gram-negative (gNeg) dataset consists of 1,444 sequences belonging to one of the five classes: cytoplasm (278), cytoplasmic membrane (309), periplasm (276), outer membrane (391) and extracellular (190). The Gram-positive (gPos) dataset consists of 541 sequences belonging to one of the four classes: cytoplasm (194), cytoplasmic membrane (103), cellwall (61) and extracellular (183).
- **TargetP datasets [30]:** The plant 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 nonPlant dataset consists of 2,738 sequences belonging to one of the three classes: mitochondrial (361), secretory pathway/signal peptide (715) and other (consisting of 1,224 proteins labeled nuclear and 438 proteins labeled cytosolic).

C. 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 [25], 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: $gPos \rightarrow gNeg$, $gNeg \rightarrow gPos$, $plant \rightarrow nonPlant$ and $nonPlant \rightarrow plant$, 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 $gPos/gNeg$ pairs, and mitochondrial, secretory pathway/signal peptide and others for $plant/nonPlant$ 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, pc -mers and k -mers. The details of how these sets of features were generated are provided below:

- For pc -mers, we invoke the proposed approach (Section III–D) 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 pc -mers. We denote the number of pc -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 pc -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, pc -mers, N_{c-mers} . For feature selection, we use *Entropy based Category Coverage Difference* (ECCD), proposed in [31], 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, when computing the feature-class dependency scores.

V. RESULTS AND DISCUSSIONS

A. Supervised Learning

This section evaluates the predictive power of pc -mers in a supervised learning scenario. Our goal is to compare the predictive power of a small number of pc -mers with the predictive power of the same number of k -mers. As discussed in Section IV–C, feature selection was used with all available labeled data in order to select top f features, from pc -mers and k -mers, respectively, that were further used to represent sequences. In order to address the second question from Section IV–A, the performance of classifiers learned from pc -mers was compared to the performance of classifiers learned from k -mers, while varying the number of features f , with $f \in \{50, 100, 150, 200, 250, 500, 1,000, 1,500\}$. We can observe

that the complexity to accuracy tradeoff with *pc*-mers is *good* (when compared to *k*-mersAll), i.e., the AUC decreases by only about 3% (in Table I, AUC when 1500 *pc*-mers were used is 0.923, while for *k*-mersAll, the AUC is 0.950) when the number of features decreases from 114,781 (average of total *k*-mers for four datasets) to 1,500.

AUC values of NBM classifiers learned for various number of features f , for *pc*-mers and *k*-mers are reported in Table I. For each dataset, maximum AUC between *pc*-mers and *k*-mers for each variation of the number of features selected is reported in **bold** font. Besides using the reduced number of features, to get an estimate of the best AUC that can be achieved using *k*-mers, AUC values of the NBM classifier learned using all the *k*-mers are also reported in Table I, referred to as *k*-mersAll. The *pc*-mers outperformed *k*-mers in 29 out of 32 experiments, whereas the *k*-mers outperformed the *pc*-mers in three experiments, corresponding to nonPlant dataset. As described in Section IV–B, the nonPlant data is skewed towards the third class (nuclear+cytosolic) as opposed to other datasets. Therefore, we hypothesize that for the nonPlant dataset, most of the R samples potentially had more than half of the sequences belonging to the third class, possibly resulting in motifs that were biased towards the third class. As a result, *pc*-mers primarily captured information corresponding to the third class, thereby explaining why *k*-mers were better than *pc*-mers for nonPlant dataset.

Overall, the results suggest that the proposed approach successfully retained informative features while reducing dimensionality by a large extent. Although the set of *k*-mers is a superset for the set of *pc*-mers, when feature selection was applied, the kept *k*-mers were not as informative as the *pc*-mers. Subsequences responsible for a particular biological problem are believed to possibly span different sequences belonging to the same class of problems, with possible mismatches. Our approach was used to primarily identify subsequences that were similar to each other based on substitution score. Therefore, *pc*-mers contained a small set of highly informative features, and when feature selection was applied on top of this set of features, feature class dependency scores were computed more precisely because all feature class dependency scores were normalized by maximum Shannon Entropy of all features, which, in the case of *pc*-mers, is limited to only informative feature variations.

B. Semi-supervised Learning

As discussed in Section IV–C, the available training data was split into various combinations of labeled and unlabeled data. For each combination of labeled and unlabeled data, a co-training classifier was learned for all the four datasets (GP, GN, P, and NP) represented using *pc*-mers and *k*-mers; *k*-mers were reduced to the number of *pc*-mers using feature selection. In order to address the fifth question from Section IV–A, the parameters l , d , x , q , t , R , and S were fixed to their default values, the amount of labeled data was fixed to 20%, and the amount of unlabeled data was varied, using 20%, 40%, 60%, or 80%. These results show that *pc*-mers outperformed *k*-mers in all cases, as shown in Table II (b), suggesting that, with small amounts of labeled data, feature selection was unable to accurately capture feature-class dependencies, thereby selecting a set of uninformative features.

C. Domain Adaptation

Table II (c) shows the AUC values for the domain adaptation algorithm trained on four source and target configurations associated with all the four datasets (GP, GN, P, and NP), using varying amount of target unlabeled data and pc -mers or k -mers as features. The amount of labeled data is fixed to 20%, and the amount of unlabeled data is varied from 20% to 80% with increments of 20%. For each comparison, the largest AUC value between pc -mers and k -mers is highlighted with bold font. In 15 out of 16 cases, the classifier had higher AUC when using pc -mer features as compared to using k -mer features, suggesting that pc -mers, by themselves, capture more informative features than k -mers, in order to predict data from the target domain.

D. Varying the Number of Motifs

Increasing the number of motifs selected, t , adds more motifs with less significant score, potentially affecting classifier performance. However, for very small t , the total number of L -mers obtained from the selected motifs might not be sufficient to carry enough information required to learn a good classifier. Therefore, classifier performance is expected to initially increase with t and later decrease. For the variations of t considered in our experiments, no decrease in the performance of the classifier was observed with increased t . However, the rate of increase was very small for larger values of t , as shown in Table III, suggesting that increasing t may maintain or decrease the performance, thereby answering the sixth question from Section IV–A.

E. Varying the Number of Samples and Sample Size

In order to address the seventh question of Section IV–A, the parameters l , d , x , p , f , and S were fixed and R was varied as $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ in a supervised learning scenario, as discussed in Section IV–C. Although the parameters l , d , x , p , and f were set to their default values, parameter S was chosen to have different values from $\{10, 20, 30, 40, 50\}$. Figure 1 plots R (x-axis) versus AUC values (y-axis) for various values of S (curves in each plot) on all six datasets.

Varying the number of samples, R : Plots in Figure 1, show that for most values of S and datasets, AUC values increased with an increase in R , suggesting that increasing the number of samples resulted in more informative motifs. However, the slope of the plots obtained for a particular value of S tends to 0 for larger values of R , specifically for R greater than 50, suggesting that performance does not rapidly increase with larger values of R . Therefore, in most cases, a good performance can be attained with a relatively small value of R (e.g., 50).

Varying the number of sequences per sample, S : As shown in the plots in Figure 1, for almost all datasets (except nonPlant dataset), AUC values of the classifiers learned from samples of 10 sequences ($S = 10$) outperformed the rest of the classifiers learned from larger samples in most cases, suggesting that the best performance was achieved for a smaller value of S . Because increasing S significantly increases the running time of the proposed approach, the best performance had very small running time compared to other sub-optimal performances.

Graphs corresponding to various values of S converged for large values of R in most cases, suggesting that the best performance was achieved for a relatively small value of R and a small value of S combination, thereby answering the seventh question from Section IV–A. As a result, features generated using the proposed approach were effective in terms of performance and efficient in terms of running time compared to TFBSGroup applied on a larger set of sequences.

F. Varying the Substitution Score Threshold

Varying threshold s will affect the number of edges in the network, potentially affecting communities obtained from the community detection algorithm. For very small values of s , the network may be too sparse resulting in very small set of pc -mers that may not be informative. Contrarily, for a very large value of s , network density may be very high, resulting in a very large set of pc -mers. As the dimensionality of pc -mers increases, the probability of outliers in the resulting set of pc -mers also increases. Therefore, when the threshold s increased, the performance of the classifier learned from the resulting set of pc -mers was expected to initially increase and then decrease. From Table IV, this behavior was observed for plant and nonPlant datasets. For gNeg and gPos datasets, the performance decreased right from the beginning, suggesting that the optimal performance for these datasets might be obtained with a much smaller value of s ($s < 1$).

VI. 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 [1], [2] for 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 sub-sequence network. The resulting pc -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 pc -mers, we have conducted experiments in supervised, semi-supervised and domain adaptation learning scenarios. In addition, we conducted experiments to evaluate the influence of some of the parameters of our algorithm: the number of motifs f selected from each subsample, the number of samples R and the size S of each sample, and the threshold value for the substitution score s . 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 resulted in improved performance as opposed to k -mers selected based on featureclass 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]. Tangirala K and Caragea D, "Community detection-based features for sequence classification," in Proc of 5th ACM BCB, 2014, pp. 559–568.
- [2]. Jia C, Carson MB, and Yu J, "A fast weak motif-finding algorithm based on community detection in graphs," BMC bioinformatics, vol. 14, no. 1, p. 227, 2013. [PubMed: 23865838]
- [3]. Blondel VD, Guillaume J-L, Lambiotte R, and Lefebvre E, "Fast unfolding of communities in large networks," Journal of Statistical Mechanics: Theory and Experiment, vol. 2008, no. 10, p. P10008, 2008.
- [4]. Caragea C, Silvescu A, and Mitra P, "Protein sequence classification using feature hashing," in BIBM, 2011, pp. 538–543.
- [5]. Sun L, Luo H, Bu D, Zhao G, Yu K, Zhang C, Liu Y, Chen R, and Zhao Y, "Utilizing sequence intrinsic composition to classify protein-coding and long non-coding transcripts," Nucleic acids research, p. gkt646, 2013.
- [6]. Kjetil Sandve G and Drabls F, "A survey of motif discovery methods in an integrated framework," Biology Direct, vol. 1, no. 1, 2006.
- [7]. Das M and Dai H-K, "A survey of DNA motif finding algorithms," BMC Bioinformatics, vol. 8, no. Suppl 7, 2007.
- [8]. Zambelli F, Pesole G, and Pavesi G, "Motif discovery and transcription factor binding sites before and after the next-generation sequencing era," Briefings in Bioinformatics, 2012.
- [9]. Bucher P, "Weight matrix descriptions of 4 eukaryotic RNA polymerase-ii promoter elements derived from 502 unrelated promoter sequences," Journal of Molecular Biology, vol. 212, pp. 563–578, 1990. [PubMed: 2329577]
- [10]. Bailey T and Elkan C, "Unsupervised learning of multiple motifs in biopolymers using expectation maximization," Machine Learning, vol. 21, no. 1–2, pp. 51–80, 1995.
- [11]. Lawrence CE, Altschul SF, Boguski MS, Liu JS, Neuwald AF, and Wootton JC, "Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment," Science (New York, N.Y.), vol. 262, no. 5131, pp. 208–214, 10 1993.
- [12]. Leibovich L, Paz I, Yakhini Z, and Mandel-Gutfreund Y, "DRIMust: a web server for discovering rank imbalanced motifs using suffix trees," Nucleic Acids Research, vol. 41, no. W1, pp. W174–W179, 2013. [PubMed: 23685432]
- [13]. Eskin E and Pevzner PA, "Finding composite regulatory patterns in DNA sequences," Bioinformatics, vol. 18, no. suppl 1, pp. S354–S363, 2002. [PubMed: 12169566]
- [14]. Boucher C and King J, "Fast motif recognition via application of statistical thresholds," BMC Bioinformatics, vol. 11, no. Suppl 1, 2010.
- [15]. Herndon N, Tangirala K, and Caragea D, "Predicting Protein Localization Using a Domain Adaptation Naive Bayes Classifier with Burrows Wheeler Transform Features," in Proc of 6th IEEE BIBM, 2014, pp. 501–504.
- [16]. Girvan M and Newman ME, "Community structure in social and biological networks," Proc of the national academy of sciences, vol. 99, no. 12, pp. 7821–7826, 2002.
- [17]. Newman ME and Girvan M, "Finding and evaluating community structure in networks," Physical review E, vol. 69, no. 2, p. 026113, 2004.
- [18]. Clauset A, Newman ME, and Moore C, "Finding community structure in very large networks," Physical review E, vol. 70, no. 6, p. 066111, 2004.

- [19]. Guimera R, Sales-Pardo M, and Amaral LAN, "Modularity from fluctuations in random graphs and complex networks," *Physical Review E*, vol. 70, no. 2, p. 025101, 2004.
- [20]. Medus A, Acuna G, and Dorso C, "Detection of community structures in networks via global optimization," *Physica A: Statistical Mechanics and its Applications*, vol. 358, no. 2, pp. 593–604, 2005.
- [21]. Nan D and Zhang X, "Prediction of hot regions in protein-protein interactions based on complex network and community detection," in *IEEE BIBM*, 2013, pp. 17–23.
- [22]. Mahmoud H, Masulli F, Rovetta S, and Russo G, "Community detection in protein-protein interaction networks using spectral and graph approaches," in *Computational Intelligence Methods for Bioinformatics and Biostatistics*. Springer, 2014, pp. 62–75.
- [23]. Mallek S, Boukhris I, and Elouedi Z, "Predicting proteins functional family: A graph-based similarity derived from community detection," in *Intelligent Systems' 2014*. Springer, 2015, pp. 629–639.
- [24]. Tangirala K, Herndon N, and Caragea D, "Community detection-based feature construction for protein sequence classification," in *Proc of 11th ISBRA*, 2015.
- [25]. Herndon N and Caragea D, "Naïve bayes domain adaptation for biological sequences," in *Proc of the 4th BIOINFORMATICS*, 2013, pp. 62–70.
- [26]. Cheng BYM, Carbonell JG, and Klein-Seetharaman J, "Protein classification based on text document classification techniques," *Proteins: Structure, Function, and Bioinformatics*, vol. 58, no. 4, pp. 955–970, 2005.
- [27]. Yang Y, Lu B-L, and Yang W-Y, "Classification of protein sequences based on word segmentation methods," in *Proc of 6th APBC*, vol. 6, 2008, pp. 177–186.
- [28]. Dayhoff MO and Schwartz RM, "A model of evolutionary change in proteins," in *In Atlas of protein sequence and structure*. Citeseer, 1978.
- [29]. Gardy JL, Laird MR, Chen F, Rey S, Walsh C, Ester M, and Brinkman FS, "Psortb v. 2.0: expanded prediction of bacterial protein subcellular localization and insights gained from comparative proteome analysis," *Bioinformatics*, vol. 21, no. 5, pp. 617–623, 2005. [PubMed: 15501914]
- [30]. Emanuelsson O, Nielsen H, Brunak S, and von Heijne G, "Predicting subcellular localization of proteins based on their n-terminal amino acid sequence," *Journal of molecular biology*, vol. 300, no. 4, pp. 1005–1016, 2000. [PubMed: 10891285]
- [31]. Langeron C, Moulin C, and Gery M, "Entropy based feature selection for text categorization," in *Proc of 2011 ACM SAC*, 2011, pp. 924–928.

Fig. 1:

Variation of AUC values on test data when sample size S was varied from 10 to 50 with increments of 10, and number of samples R , was varied from 10 to 100 with increments of 10, on all four datasets (GP, GN, P, and NP). The number of features used to represent the sequences, f , was set to 2000.

TABLE I:

Variation of the performance with the number of *pc*-mers and *k*-mers features selected for NBM classifier in supervised learning scenario. Table includes AUC values for gPos, gNeg, plant, and nonPlant datasets, averaged over the five test folds. For each dataset, maximum AUC between *pc*-mers and *k*-mers for each variation of the number of features selected is reported in bold font. Besides using the reduced number of features, to get an estimate of the best AUC that can be achieved using *k*-mers, AUC values of the NBM classifier learned using all the *k*-mers (denoted by *k*-mersAll) are also presented.

Target	Features	<i>f</i>							
		50	100	150	200	250	500	1,000	1,500
gPos	<i>pc</i> -mers	0.746	0.792	0.812	0.824	0.826	0.876	0.912	0.923
	<i>k</i> -mers	0.649	0.698	0.736	0.770	0.786	0.838	0.860	0.870
	<i>k</i> -mersAll	0.950							
gNeg	<i>pc</i> -mers	0.770	0.841	0.857	0.869	0.876	0.904	0.918	0.925
	<i>k</i> -mers	0.625	0.667	0.698	0.711	0.744	0.809	0.884	0.900
	<i>k</i> -mersAll	0.955							
plant	<i>pc</i> -mers	0.650	0.707	0.737	0.743	0.748	0.784	0.823	0.836
	<i>k</i> -mers	0.536	0.566	0.578	0.594	0.598	0.651	0.723	0.757
	<i>k</i> -mersAll	0.843							
nonPlant	<i>pc</i> -mers	0.707	0.759	0.775	0.787	0.793	0.807	0.825	0.833
	<i>k</i> -mers	0.655	0.700	0.734	0.759	0.788	0.829	0.856	0.871
	<i>k</i> -mersAll	0.868							

TABLE II:

AUC values on the four datasets: Gram-positive (gPos), Gram-negative (gNeg), plant (plant), and non-plant (nonPlant) - averaged over the five test folds. 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 *pc*-mers.

(a) Supervised								
Unlabeled (%)	Features							
	<i>pc</i> -mers	<i>k</i> -mers	<i>pc</i> -mers	<i>k</i> -mers	<i>pc</i> -mers	<i>k</i> -mers	<i>pc</i> -mers	<i>k</i> -mers
	Target							
	gPos		gNeg		plant		nonPlant	
0	0.925	0.869	0.929	0.915	0.837	0.754	0.834	0.874
(b) Semi-supervised								
Unlabeled (%)	Features							
	<i>pc</i> -mers	<i>k</i> -mers	<i>pc</i> -mers	<i>k</i> -mers	<i>pc</i> -mers	<i>k</i> -mers	<i>pc</i> -mers	<i>k</i> -mers
	Target							
	gPos		gNeg		plant		nonPlant	
20	0.847	0.746	0.877	0.793	0.720	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	0.870	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	0.770
(c) Domain adaptation								
Unlabeled (%)	Features							
	<i>pc</i> -mers	<i>k</i> -mers	<i>pc</i> -mers	<i>k</i> -mers	<i>pc</i> -mers	<i>k</i> -mers	<i>pc</i> -mers	<i>k</i> -mers
	Target							
	gNeg→gPos		gPos→gNeg		nonPlant→plant		plant→nonPlant	
20	0.877	0.785	0.911	0.899	0.802	0.780	0.829	0.763
40	0.856	0.755	0.910	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	0.730
80	0.839	0.741	0.892	0.895	0.734	0.727	0.803	0.744

TABLE III:

AUC values obtained on test data with NBM classifiers learned in the supervised scenario, when top t motifs are used as features. The number of motifs, t , was varied from 1 to 50. In addition to performance, we also show the total number of pc -mers obtained for a set of t motifs ($\#pc$ -mers), averaged over five folds.

t	gPos		gNeg		plant		nonPlant	
	$\#pc$ -mers	AUC	$\#pc$ -mers	AUC	$\#pc$ -mers	AUC	$\#pc$ -mers	AUC
1	152	0.824	147	0.789	145	0.691	146	0.704
5	867	0.905	805	0.901	800	0.792	765	0.797
10	1,976	0.925	1,822	0.929	1,684	0.837	1,750	0.834
15	3,158	0.936	2,936	0.939	2,593	0.854	2,888	0.857
25	5,506	0.944	5,139	0.951	4,361	0.868	5,357	0.880
35	7,700	0.945	7,237	0.955	6,097	0.876	7,884	0.892
50	10,522	0.948	9,800	0.956	8,444	0.875	11,323	0.898

TABLE IV:

AUC values obtained on test data with NBM classifiers learned in a supervised scenario, for protein motifs of lengths {2, 3, 4}, when varying the substitution score used when constructing the network from 1:5:10:15 (shown on columns).

Target	s			
	1	5	10	15
gPos	0.882	0.888	0.913	0.925
gNeg	0.937	0.941	0.938	0.929
plant	0.833	0.841	0.840	0.837
nonPlant	0.871	0.872	0.868	0.834