Discriminative Prediction of Enhancers with Word Combinations as Features

Pham Viet Hung and Tu Minh Phuong

Abstract. Identification of enhancer regions is important for understanding the regulation mechanism of gene expression. Recent studies show that it is possible to predict enhancers using discriminative classifiers with generic sequence features such as k-mers or words. The accuracy of such discriminative prediction largely depends on the ability of the models to capture not only the presence of predictive k-mers (words), but also spatial constraints on clusters of such k-mers. In this paper, we propose a method that first selects the most important word features and then use combinations of such words, which satisfy certain spatial constraints, as additional features. Experiments with real data sets show that the proposed method compares favorably with a state-of-the-art enhancer prediction method in terms of prediction accuracy.

Keywords: Enhancer prediction, SVM, feature extraction, TFBS combination.

1 Introduction

The regulation of gene expression plays a fundamental role in cell differentiation and responses of cells to various conditions. There are several levels, at which the expression of genes is regulated, the most important of which is transcriptional regulation. At the transcriptional level, the expression of genes is regulated by transcriptional factors (TFs) that recognize and bind to short DNA sequence motifs, known as transcription factor binding sites (TFBSs). To provide stronger signals for TFs, TFBSs often occur near each other in DNA regions, which are called *cis*-regulatory modules (CRM). CRMs that enhance the expression of genes from distance are called *enhancers*. Identification of enhancers is important for understanding the mechanisms of gene expression regulation.

Pham Viet Hung · Tu Minh Phuong
Department of Computer Science,
Posts and Telecommunications Institute of Technology, Hanoi, Vietnam
e-mail: {hungpv,phuongtm}@ptit.edu.vn

V.-H. Nguyen et al. (eds.), *Knowledge and Systems Engineering,*
Advances in Intelligent Systems and Computing 326, DOI: 10.1007/978-3-319-11680-8_4

With the current technologies, laboratory methods for enhancer identification are available. Usually, this consists of two steps. First, chromatin immunoprecipitation (ChIP) technique is used to detect signatures of specific TFs associated with activities of enhancers. Then, microarray hybridization or massive parallel sequencing is used to decode the enhancers involved in these activities [14]. This approach allows recognition of enhancers with high accuracy but is resource-intensive, thus limiting its use in practice. A faster and more cost-effective approach is to use computational techniques for predicting enhancers [15], which is the focus of this paper. Methods of this type take advantage of the availability of sequence and other genomic data to recognize enhancers.

A large group of enhancer prediction methods rely on analysis of sequence data. Methods of this group mainly use two strategies. The first strategy is to use predetermined TFBSs, for example from TFBS databases or by running a motif finding algorithm, and search for clusters of these TFBSs [1]. These methods depend on the availability of known TFs and their motifs. The second strategy is discriminative, i.e. using classification algorithms with confirmed enhancers as training data to differentiate between enhancers of a specific type and non-enhancers [12, 7].

Sequence-based enhancer prediction approaches rely on the assumption that similar sequence content is associated with similar binding events, which are in turn associated with similar gene expression. It is well known that TF binding is sequence-specific, i.e. each TF recognizes and binds to a specific short DNA motif (TFBS) of length up to 10 base-pairs (bp). It is also observed that the presence of a motif is not a guarantee for binding and a large fraction of motifs are false positives, i.e. are not associated with binding events. For binding events to occur, TFBSs usually cluster together in enhancers to provide stronger signals for TFs. Previous studies show that there are certain constraints on motif types, motif numbers, and relative motif location within those clusters [5, 14]. For example, pair or triple of TFs of certain types have been observed to bind to closely located TFBSs to coregulate the expression of some genes [18]. Thus, to successfully predict enhancers, classifier-based methods should be able to model such constraints.

In this paper, we follow the classification-based approach and use SVM classifiers to discriminate between enhancer and non-enhancer sequences. Our method is similar to one presented in [12, 7, 16], which also use SVM for this problem. The difference is that we introduce a new type of features that explicitly count for the constraints on types, distance and order of combinations of words (k-mers). To make counting of word combinations tractable, we use feature selection to reduce the set of words to consider. We develop an algorithm to count the number of word combinations that satisfy certain spatial constraints on the locations of words. In this paper, only word pairs are considered but experiments show that this leads to improvement in prediction accuracy in many cases. In experiments using several enhancer datasets from three species (human, mouse, and *Caenorhabditis elegans*), our method compares favorably with a state-of-the art enhancer prediction method. An analysis of k-mer pairs extracted from the models learned by the method also provides interesting patterns of TFBS clusters from experimented datasets. Importantly, this is achieved without significant increase in computational complexity.

1.1 Related Work

A significant number of computational methods for CRM prediction, in general, and enhancer prediction, in particular, have been developed. Some methods rely on the assumption that CRMs, as a functional regions, are more conserved between related species than background, non-functional regions. These methods search for conserved regions as putative CRMs by combining phylogenetic footprinting with sequence information [11]. For certain types of CRMs, this conservation-based approach can make predictions with high accuracy. However, it is known that many CRMs that are not highly conserved, for which this approach produces poor predictions [15].

Another group of methods rely on signals contained in genomic sequences to make predictions. Early methods of this group search for clusters of known TFBSs within a sequence window [1]. Other methods construct probabilistic models of CRMs, for example in forms of HMMs, and search for regions that fit the learned models with high probabilities. Window clustering require databases of confirmed TFBSs while probabilistic methods require only positive examples or no examples at all but tend to produce many spurious predictions.

Recently, a new, discriminative approach has been developed. Methods of this approach take as input both positive (CRM sequences) and negative examples (sequences that are believed to be not CRMs) and build a classification model to discriminate positive sequences from negative ones. Several studies have shown the ability of discriminative methods in predicting enhancers of complex organisms such as mammals, which is challenging for other approaches [7, 9, 4, 12]. Besides the availability of training data, the success of classification-based approach depends, to a large degree, on the selection of appropriate features to represent sequences, or, more generally, on the selection of appropriate similarity measures [4] and kernels [12] between sequences. While some methods of this group use confirmed motifs as features [12], thus depends on the availability of TFBS databases, other methods extract features from input sequences, making them easier to use in practice [7, 4, 16]. Our proposed method is similar to [7, 4, 16] in that it uses generic sequence features. However, we introduce additional features by explicitly counting the numbers of word combinations. In this way, our approach is able to incorporate different constraints on the presence of motifs within enhancers.

2 Preliminaries

In this work, we follow the discriminative approach to predict enhancers. Specifically, we train Support Vector Machines (SVMs) classifiers to differentiate between enhancer (positive) and non-enhancer (negative) sequences. SVMs have been the technique of choice in many enhancer prediction methods due to their superior accuracy and flexibility in dealing with different types of biological data such as sequences and interaction networks. The success of SVMs classifiers depends, to a large extent, on choosing appropriate features or kernels. In this section,

we review popular features for sequence data, which serve as basis for our new types of features.

The Spectrum and Mismatch Kernels. In general, when classifying biological sequences with SVMs, one needs to measure the similarity between each pair of sequences and use it as kernel. This process normally involves alignment mechanism of some kind, which makes the similarity computation expensive. Leslie *et al* [8] proposed a simple type of features and respective kernel for sequences that is alignment-free: the *spectrum kernel*. A sequence s of length l with alphabet α ($\alpha = \{A,C,T,G\}$ for DNA sequences) is scanned and the numbers of occurrences of words of length k, or k-mers is used to build a feature vector for s (in this paper, "word" and "k-mer" are interchangeable). For DNA sequences, this method creates a feature vector with 4^k elements corresponding to 4^k distinct k-mers. The inner product of two such feature vectors is calculated as the spectrum kernel function of corresponding sequences. This kernel is alignment-free since the similarity of two sequences could be computed without using any alignment, hence making such kernel computational efficient.

The spectrum kernel can be extended to incorporate partial matches of k-mers, which is important in comparison of sequences with less conserved motifs. Such variation of the spectrum kernel is known as *mismatch kernel*. A (k,m) mismatch kernel considers two k-mers the same if they have no more than m mismatches.

The spectrum and mismatch kernels are simple to calculate and have been shown to deliver satisfactory results in certain cases [7]. Both types of kernels measure the similarity based on the co-occurrences of k-mers in a pair of sequences independent of k-mers' positions. However, it is well known that, in many cases, for binding events to occur certain constrains on the locations, and orders of motifs (TFBS) should be met. For example, some enhancers consist of pairs or triples of instances of the same or different motifs that are located near each other (within tens of bp), and the spectrum and mismatch kernel may not work well in these situations. To model such constrains, we present novel kernels that explicitly take into account the relative locations of k-mers.

3 Word Combination Features

In this section, we introduce a new type of features and kernel that explicitly incorporate location and order constrains on occurrences of k-mers or words, which we call *word combination feature*(WCF). Basically, the values of such features are the numbers of times each pair of k-mers co-occurs within a sequence and also satisfies certain location constrains. For example, a possible feature is the number of times k-mer A co-occurs with k-mer B and the distance between the two instances are less than a predefined threshold. There are two main obstacles in using such features. First, the number of k-mer pairs is very large, resulting in very high-dimensional feature space. With $k = 6$, for example, there are thousands of possible k-mers and millions of their pairs. Second, calculating such features may be time-consuming and thus requires the development of efficient algorithms.

In what follows, we describe solutions for these two problems. Specifically, to reduce the number of k-mer pair features, we select a small set of important k-mers and compute pair features only for k-mer from this set. Then, we introduce an algorithm that calculate such features with complexity linear to the sequence length.

3.1 Selection of Important K-mers

Although there are 4^k possible k-mers, only a small number of them are important in the sense that they are predictive of an enhancer. By focusing only on these important features, we can reduce the number of k-mer pairs to a manageable size. In this work, we use two methods to select important k-mers based on feature selection ability of linear SVMs and AdaBoost [3].

Selecting Important k-mers with SVMs. A linear SVM uses a linear score function of the form $f(x) = \sum_{i=0}^{N} w_i x_i$ to calculate a score, which is then thresholded to decide the class label. In this function, x_i is i-th feature and w_i is its weight learned from training data. The larger the absolute value of w_i, the higher contribution of i-th feature to the score function, and thus the more important it is. Our selection method works as follows. First, we train a SVM using the spectrum kernel, i.e. with all k-mers. Once the SVM has been trained, we sort k-mers based on their weights, and then use three strategies to select most important ones: 1) select k-mers with top positive weights (SVM+), 2) k-mers with top negative weights (SVM−), and 3) combined list of k-mers with top positive and negative weights (SVM+−). Here, "top negative weights" mean negative weights with highest absolute values.

Selecting Important k-mers with AdaBoost. AdaBoost [3] is a special case of boosting algorithms. In general, boosting works by combining many weak classifiers (each has a slightly better prediction accuracy than choosing at random) to produce a strong classifier. Each of these weak classifiers could be as simple as a decision stump model. The method of learning is an iterative process of growing an ensemble of weak classifiers, each time adding one more. AdaBoost is adaptive since subsequent classifier added are selected to focus on examples mis-classified by previous classifiers.

In this work, we use AdaBoost with decision stumps as weak classifiers. A decision stump $ds(i, \theta)$ is an one-level decision tree, which has the form "*class = positive if* $i \geq \theta$*; and class = negative otherwise*". In each iteration, AdaBoost selects i and θ so that the most number of training samples are classified correctly. Therefore, the algorithm tends to select most predictive features in early iterations, and multiple times. We use this property to select predictive k-mers as follows. We train AdaBoost on the training data using all k-mers as features. Each time a k-mer is selected in a decision stump, it is added into the set of important k-mers. This process ends when a desired number of distinct k-mers has been selected.

3.2 Calculating Combination Features

Now, we describe an efficient algorithm to extract features that are combinations of important *k*-mers selected in the previous step. Recall that we use features that are numbers of times pairs of important *k*-mers or two instances of a same *k*-mer occur within a predefined distance d. When counting these features, we do not consider order of words, and we do not differentiate between a word and its reverse complement. In other words, pairs of *k*-mers A and B with A appear before B or B before A or pair of A and reverse complement of B will all be counted as one feature.

To count the number of such pairs, first we map each important *k*-mer and its reverse complement to an unique index. For example, 100 *k*-mers will be assigned indexes from 1 to 100 while their reverse complements will be assigned indexes from -1 to -100. This step produces a *k*-mers dictionary called *k_mers_dict* that allows our algorithm to compress the DNA sequence to an array of indexes so that comparison will be faster.

The second preprocessing step produces another dictionary allowing the feature extraction step to accurately detect each feature. In this step, we determine the set of index pairs by calculating all possible pair combinations of *k*-mers indexes. Then, combinations corresponding to the same feature are then bagged to form the features mapping.

This second step produces a feature index dictionary called *feat_index_dict* that mapping pairs of *k*-mer indexes to feature indexes. Each genomic sequence is then processed to extract features by using the algorithm presented in Algorithm 1.

In the worse case scenario we will have to analyze at most $(L-k)*(D+k)$ *k*-mers indexes with L being the length of DNA sequence and D is the maximum distance of *k*-mers pair.

Combining Features. Once combination features are calculated, we remove features that does not appear in any training sample. The remaining features are then normalized to sum to one and the resulting feature vector is concatenated to spectrum kernel's feature vector to form a new feature vector. Note that, the two set of features are normalized separately. We then normalize again after combining these features so they contribute equally to the result. For any two sequences, the inner product of their feature vectors forms the kernel value.

4 Experiments and Results

4.1 Data and Settings

4.1.1 Datasets

We evaluated the effectiveness of the proposed method on datasets containing enhancers for multiple TFs and their co-factors as well as histone marks from human,

```
Data: D, k, k_mers_dict, feat_index_dict and seq
Result: feat_vector
scan seq to produce k-mers list k_mers_list;
for i = 0 to length(k_mers_list) do
    kmer = k_mers_list[i];
    find index ki for kmer in k_mers_dict;
    index_array[i] = ki;
end
for i = 0 to length(index_array) do
    idx1 = index_array[i];
    if idx1 ≠ 0 then
        for j = 0 to i + D + k do
            idx2 = index_array[j];
            if idx2 ≠ 0 then
                find index fi for idx1 : idx2 in feat_index_dict;
                feat_vector[fi] + +;
            end
        end
    end
end
```

Algorithm 1. Algorithm for counting word combination features

mouse, and C. elegans. Specifically, as positive datasets, we used datasets provided by Yanez-Cuba *et al* [18] and Fletez-Brant *et al* [2].

Yanez-Cuba *et al* [18] have compiled several sets of enhancers based on data from previous work. The datasets include enhancers for the following TFs and histone marks: TAL1 (from [10]), HNF4A,GATA6, CDX2, and H3K4me2 (from [17]), for different cell lines. The original datasets contain only short fragments corresponding to ChIP peaks. Therefore, for each fragment, we extended from 300 to 500 bp in both directions to get an enhancer sequence of approximately 1000 bp, and used these sequences to form positive sets. To generate negative sets, we use the method by Lee *et al* [7]. This method generates null sequences (without enhancers) by randomly selecting DNA sequences from the same genome and have the same length and repeat fraction distributions. The Kmer-Svm Server [2] implements this method and we use this server to generate negative datasets for all our experiments. In the following section, we will use this collection of datasets (referred to as the *first collection*) for exploring different settings of our method as well as for comparison with existing methods.

The second collection of datasets is the same as used by Fletez-Brant *et al* [2]. These datasets contain enhancers obtained through ChIP-seq or DNase-seq experiments for several TF binding sites: ESRRB (in mouse ES cells), GR (in mouse 3134 and AtT20 cells), EWS-FLI (in human EWS502 and HUVEC cells). The datasets were provided with both positive and negative sequences, extended from peak

fragments appropriately and we used them as is. In our experiments, we used the datasets from this second collection only for comparison of our and other methods.

4.1.2 Experiment Settings

We implemented the proposed method in Matlab using the built-in SVM algorithm with linear kernel and parameter $C = 1$. All experiments were performed with hexamers ($k = 6$). This value of k has been proved to deliver the best performance [7]. For each set of data, we ran several experiments, varying the following parameters: method of selecting k-mers, number of selected k-mers, maximum distance between two k-mers in a valid feature.

The performance of the classifier was judged by two metrics: the area under the ROC curve (AUROC) and the PR curve (AUPRC). The AUROC is the area under the ROC which is a curve plotting true positive rate (sensitivity) against false positive rate (1-specificity) at different SVM score thresholds. It measures the probability that a randomly selected positive sample will score higher than a randomly selected negative sample. The PR curve plots Precision against Recall and AUPRC could be interpreted as what is the probability that a sequence really contains enhancer if the classification said so. The ROC could yield a better performance for a classifier in the case of imbalance training examples where as the PR curve directly assesses the accuracy of positive predictions.

All classifiers were evaluated with 5-fold cross validation protocol, in which a classifier is trained with four fifth of the data set and tested on the rest. The AUC and PR scores are averaged over the five folds.

4.2 Results

4.2.1 The Effect of Feature Selection Methods

The first experiment was designed to verify the effect of feature selection methods on prediction accuracy. Recall that for this experiment, we used only datasets from the first collection. The SVM−based and AdaBoost-based feature selection method were run to select 100 most important hexamers. For the SVM−based method, all three strategies were used. Selected hexamers were then used to produce combination features with distance between two hexamers not exceeding 100 bp. Table 1 summarizes AUROC and AUPRC scores for five datasets using the four selection methods. In all tables, (j), (p) and (d) stand for jurkat, proliferating and differentiated cell lines respectively. As shown, the selection method with the best AUC values is SVM+. SVM+ achieved the highest AUROC scores in four and the best AUPRC scores in three out of five datasets. The second best SVM+− achieved the highest scores in just one case. The results also show that using k-mers with negative weights as features is harmful for prediction accuracy. The SVM− option, i.e. using only negative weight k-mers, achieved the worst accuracy. In general, features selected by SVM yield higher AUC scores than by AdaBoost, making it more convenient to use SVM for both feature selection and subsequent classification.

Table 1 AUC scores for different feature selection methods

Method	**SVM+**		**SVM+−**		**SVM−**		**AdaBoost**	
	ROC	PR	ROC	PR	ROC	PR	ROC	PR
TAL1(j)	**0.9133**	**0.5846**	0.9088	0.5829	0.8302	0.3847	0.9046	0.5602
HNF4A(d)	0.8395	0.3903	**0.8416**	**0.4002**	0.7748	0.2899	0.8368	0.3811
GATA6(p)	**0.9754**	**0.8218**	0.9734	0.8161	0.8992	0.5774	0.9726	0.8113
CDX2(d)	**0.8594**	**0.4254**	0.8547	0.4222	0.7936	0.3338	0.8492	0.4190
H3K4me2(d)	**0.7976**	0.2952	0.7959	0.2996	0.7721	0.2633	0.7966	**0.3068**

4.2.2 The Effect of Feature Number and Distance

In the second experiment, we used SVM+, the method that has delivered the most accurate results, and experimented with different feature numbers and distance values. The number of selected k-mers (N) was set to 10, 30, 50, 100, and the maximum distance (D) between two k-mers was set to 10, 30, 50, 100, and 200 bp. Due to space limit, Tab. 2 show only highest AUC values and corresponding N and D.

As shown, $N = 100$ yielded superior AUPRC scores for all five datasets, although it achieved the best AUROC scores for only two out of five datasets. The best AUROC scores for the other three datasets were achieved with $N = 30$, although the difference in AUROC scores for N=30, 50, and 100 is not statistically significant (according to paired T-tests with threshold 0.05). The best AUROC and AUPRC scores were achieved with $D = 100$ on three datasets and $D = 30$ on two other datasets. The difference in the best values of $D = 100$ may be attributed to the variability in spatial constraints for different enhancer types, as reported previously [18]. Overal, the combination of $N = 100$ and $D = 100$ provides the best results and will be used in the remaining experiments.

Table 2 The best AUROC and AUPRC scores and corresponding N and D for SVM+ feature selection method

Dataset	**Best AUROC**	**N**	**D**	**Best AUPRC**	**N**	**D**
TAL1(j)	0.9133	100	100	0.5846	100	100
HNF4A(d)	0.8518	50	200	0.4141	50	200
GATA6(p)	0.9769	30	100	0.8265	50	200
CDX2(d)	0.8594	100	100	0.4301	50	200
H3K4me2(d)	0.8050	30	30	0.3109	10	200

4.2.3 Comparison with Existing Methods

We compared our method with the method by Lee *et al* [7] (referred to as SK). In that method, linear SVMs with spectrum kernels are used to predict enhancers from genomic sequences. Experiments have shown that the method achieves state-of-the-art prediction accuracy in predicting mouse enhancers [7, 2] and therefore we used only the method by Lee *et al* in our comparison. The authors of that method provides an implementation of the method at http://kmersvm.beerlab.org, which we used with default parameters in our experiments. The size of negative sets was set to 10000 sequences. The comparison was performed in the datasets from both collections. Our method (called WCF) was run with SVM+, $N = 100$, and $D = 100$.

Table 3 summarizes the average AUROC and AUPRC scores of the methods. For all the datasets from the first collection, the proposed method outperforms SK in terms of both AUROC and AUPRC. The improvement of AUROC is from nearly 2% (for H3K4me2(d)) to 5%(TAL1(j)). The improvement of AUPRC is more substantial, which is more than 10% in two cases (TAL1(j) and GATA6(p)). For the datasets from the second collection, two methods achieve comparable results. More precisely, the proposed method perform worse than SK in three out of five datasets, however the differences are negligible ($< 0.5\%$) and not statistically significant (paired t-test). A possible explanation for the difference in performance of our method with two data collections is the differences in organization of TF binding sites in two cases. Since our method explicitly model the constraints on relative locations of combinative binding sites, it would be more suitable for enhancers with such constraints, which seem to be the first case, while it does not influence the results when such constraints are not tight, and the second collection may be such a case. Overall, our method outperforms SK substantially in half the cases and performs comparably in the other cases.

4.2.4 Analysis of Important Features

After training, a linear SVM outputs a weight vector, each element of which corresponds to an input feature. Features with larger absolute weights are more important because they contribute more to the final score. Following Lee *et al* [7], we ask if features with large positive weights are also biologically meaningful. For each dataset, we list the features, both single and combination, with the highest positive weights and find corresponding (if any) TFs in databases of known TFBSs. Due to space limit, we show only top 10 features for CDX2(d) (Tab. 4). As shown, nine out of 10 top features are combination ones, suggesting that for this dataset, combination features are more predictive than single ones. More importantly, most highly ranked words are known TFBSs for CDX2, GATA, HNF4A, FOXA, AP-1, suggesting that combinations of these motifs are important for CDX2 binding events to occur. To verify this, we compare the results with previous findings. Using experimental methods, Verzi *et al* [17] found that CDX2 TF partners with distinct motifs during different cell states. Specifically, GATA motifs are found close to the binding site of CDX2 during proliferating state while CDX2 binding site regions specific to

Table 3 Comparison of enhancer prediction with Spectrum Kernel (SK) [7] and Word Combination Feature Kernel (WCF)

Kernel	**WCF**		**SK**		**Differences**	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
TAL1(j)	0.9133	0.5846	0.8678	0.4785	+0.0455	+0.1062
HNF4A(d)	0.8395	0.3903	0.8164	0.3526	+0.0231	+0.0377
GATA6(p)	0.9754	0.8218	0.9499	0.7180	+0.0255	+0.1038
CDX2(d)	0.8594	0.4254	0.8276	0.3895	+0.0318	+0.0359
H3K4me2(d)	0.7976	0.2952	0.7849	0.2902	+0.0128	+0.0050
EWS502	0.9612	0.9527	0.9640	0.9570	-0.0028	-0.0043
HUVEC	0.9621	0.9610	0.9600	0.9590	+0.0021	+0.0020
3134	0.8934	0.8701	0.8970	0.8740	-0.0036	-0.0039
Att20	0.9051	0.7769	0.9050	0.7840	+0.0001	-0.0071
ESRRB	0.9148	0.9282	0.9160	0.9310	-0.0012	-0.0028

Table 4 Top 10 features with highest positive weights as returned by SVM for CDX2 (d) binding sites

Features		**Reverse complements**		**SVM weights**	**Known TF(s)**	
CATAAA	CTTATC	TTTATG	GATAAG	15.804	CDX2	GATA
AGGGCA	CATAAA	TGCCCT	TTTATG	14.891	HNF4A	CDX2
CAAACA	CAAAGG	TGTTTG	CCTTTG	14.613	FOXA	HNF4A
AATAAA	GACTCA	TTTATT	TGAGTC	14.590	CDX2	AP-1
ATAAAA	CTTATC	TTTTAT	GATAAG	14.394	CDX2	GATA
GCCCCA	GGCCCC	TGGGGC	GGGGCC	13.826		
AGAGAG		CTCTCT		13.471	GATA	
AGTCAT	CATAAA	ATGACT	TTTATG	13.053	AP-1	CDX2
CAAAGG	TAAACA	CCTTTG	TGTTTA	13.007	HNF4A	CDX2
CATAAA	CCACCC	TTTATG	GGGTGG	12.905	CDX2	

differentiated cell show a significant enrichment of HNF4A, AP-1 and FOXA motifs. These are almost the same motifs we found. The agreement between motif sets found by our method and reported by Verzi *et al* provides evidence that the motifs of highly ranked combination feature are biologically meaningful, and the proposed method can be used to get insight of enhancer organization.

5 Conclusion

We have presented a novel method for enhancer prediction using only sequence data. Based on generic features in forms of words extracted from genomic sequences, we introduce a new type of features that are pairs of words satisfying certain constraints on their locations. We have developed a fast feature extraction method that combines feature selection with a fast word pair counting algorithm. In a comparison with a leading method, using such word pairs as additional features for SVM classifiers has resulted in improvements of prediction accuracy as measured by AUC values of ROC and PR in half the cases while does not affect the accuracy in the others. The most important word combination features found by SVMs are biologically meaningful, thus providing additional information about enhancer content and structure. In this work, we consider only pairs of words and one type of spatial constraints (distance). However, the method can be extended to consider other types of constraints as well as combinations with more than two words to cover cases with more complex enhancer organization.

References

[1] Bailey, T.L., Noble, W.S.: Searching for statistically significant regulatory modules. Bioinformatics 19(suppl. 2), ii16–ii25 (2003)
[2] Fletez-Brant, C., Lee, D., McCallion, A.S., Beer, M.A.: kmer-SVM: a web server for identifying predictive regulatory sequence features in genomic data sets. Nucleic Acids Res. 41(Web Server issue), W544–W556 (2013)
[3] Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences 55(1) (1997)
[4] Göke, J., Schulz, M.H., Lasserre, J., Vingron, M.: Estimation of Pairwise Sequence Similarity of Mammalian Enhancers with Word Neighbourhood Counts. Bioinformatics 28(5), 656–663 (2012)
[5] Kim, T., Hemberg, M., Gray, J.M., Costa, A.M., Bear, D.M., Wu, J., Harmin, D.A., Laptewicz, M., Barbara-Haley, K., Kuersten, S., et al.: Widespread transcription at neuronal activity-regulated enhancers. Nature 465, 182–187 (2010)
[6] Langmead, B., Trapnell, C., Pop, M., Salzberg, S.L.: Ultrafast and memoryefficient alignment of short DNA sequences to the human genome. Gen. Biol. 10, R25 (2009)
[7] Lee, D., Karchin, R., Beer, M.A.: Discriminative prediction of mammalian enhancers from DNA sequence. Gen. Res. 21(12), 2167–2180 (2011)
[8] Leslie, C., Eskin, E., Noble, W.S.: The spectrum kernel: A string kernel for SVM protein classification. In: Proc. of Pac. Symp. Biocomput. 2002 (2002)
[9] Leung, G., Eisen, M.B.: Identifying cis-regulatory sequences by word profile similarity. PLoS One 4, e6901 (2009), doi:10.1371/journal.pone.0006901
[10] Palii, C.G., Perez-Iratxeta, C., Yao, Z., Cao, Y., Dai, F., Davison, J., Atkins, H., Allan, D., Dilworth, F.J., Gentleman, R., et al.: Differential genomic targeting of the transcription factor TAL1 in alternate haematopoietic lineages. EMBO J. 30, 494–509 (2011)
[11] Pierstorff, N., Bergman, C.M., Wiehe, T.: Identifying cis-regulatory modules by combining comparative and compositional analysis of DNA. Bioinformatics 22, 2858–2864 (2006)

[12] Schultheiss, S.J., Busch, W., Lohmann, J.U., Kohlbacher, O., Ratsch, G.: KIRMES: Kernel-based identification of regulatory modules in euchromatic sequences. Bioinformatics 25(16), 2126–2133 (2009)
[13] Sinha, S., He, X.: MORPH: probabilistic alignment combined with hidden Markov models of cis-regulatory modules. PLoS Comput. Biol. 3, e216 (2007)
[14] Spitz, F., Furlong, E.E.M.: Transcription factors: from enhancer binding to developmental control. Nature Reviews Genetics 13, 613–626 (2012)
[15] Su, J., Teichmann, S.A., Down, T.A.: Assessing Computational Methods of Cis-Regulatory Module Prediction. PLoS Comput. Biol. 6(12), e1001020 (2010)
[16] Thanh, H.V., Phuong, T.M.: Enhancer Prediction Using Distance Aware Kernels. In: Proc. of RIVF 2013 (2013)
[17] Verzi, M.P., Shin, H., He, H.H., Sulahian, R., Meyer, C.A., Montgomery, R.K., Fleet, J.C., Brown, M., Liu, X.S., Shivdasani, R.A.: Differentiation-Specific Histone Modifications Reveal Dynamic Chromatin Interactions and Partners for the Intestinal Transcription Factor CDX2. Developmental Cell 19, 713–726 (2010)
[18] Yanez-Cuna, J.O., Dinh, H.Q., Kvon, E.Z.: Uncovering cis-regulatory sequence requirements for context specific transcription factor binding. Genome Research 22, 2018–2030 (2012)
[19] Zhong, M., Niu, W., Lu, Z.J., Sarov, M., Murray, J.I., Janette, J., Raha, D., Sheaffer, K.L., Lam, H.Y.K., Preston, E., et al.: Genome-wide identification of binding sites defines distinct functions for Caenorhabditis elegans PHA-4/FOXA in development and environmental response. PLoS Genet. 6, e1000848 (2010)